UNIVERSITY OF TECHNOLOGY SYDNEY

Doctoral Thesis

Multi-Author Document Decomposition Based on Authorship

Author:
Khaled Aldebei

Principal Supervisor:
Prof. Xiangjian He

Co-Supervisor:
Dr. Wenjing Jia

Co-Supervisor:
Dr. Gengfa Fang

A thesis submitted in fulfilment of the requirements
for the degree of Doctor of Philosophy

in the

Global Big Data Technologies Centre
UNIVERSITY OF TECHNOLOGY SYDNEY

January 2018

Declaration of Authorship

I, Khaled Aldebei, declare that this thesis entitled, 'Multi-Author Document Decomposition Based on Authorship' and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed:

Production Note:
Signature removed prior to publication.

Date: 16/01/2018

UNIVERSITY OF TECHNOLOGY SYDNEY

Abstract

Global Big Data Technologies Centre
UNIVERSITY OF TECHNOLOGY SYDNEY

Doctor of Philosophy

Multi-Author Document Decomposition Based on Authorship

by Khaled ALDEBEI

Decomposing a document written by more than one author into sentences based on authorship is of great significance due to the increasing demand for plagiarism detection, forensic analysis, civil law (i.e., disputed copyright issues) and intelligence issues that involves disputed anonymous documents. Among the existing studies for document decomposition, some were limited by specific languages, according to topics or restricted to a document of two authors, and their accuracies have big rooms for improvement. In this thesis, we propose novel approaches for decomposition of a multi-author document written in any language disregarding to topics, based on a Naive-Bayesian model and Hidden Markov Model (HMM). The proposed approaches of the Naive-Bayesian model aim to exploit the difference in its posterior probability to improve the performance of decomposition. Two main procedures are proposed based on Naive-Bayesian model, and they are Segment Elicitation procedure and Probability Indication Procedure. The segment elicitation procedure is proposed to form a strong labeled training dataset. The probability indication procedure is developed to improve the purity of the sentence decomposition. The proposed approaches of the HMM strive to exploit the contextual correlation hidden among sentences when determining their authorships. In this thesis, it is for the first time the sequential patterns hidden among document elements is considered for such a problem. To build and learn the HMM, a new unsupervised learning method is proposed to estimate its initial parameters. The proposed frameworks do not require the availability of any information of authors or document's context other than

how many authors have contributed to writing the document. The effectiveness of the proposed algorithms is proved using benchmark datasets which are widely used for authorship analysis of documents. Furthermore, scientific papers are used to demonstrate the performance of the proposed approaches on authentic documents. Comparisons with recent state-the-art approaches are also presented to demonstrate the significance of our new ideas and the superior performance of the proposed approaches.

Acknowledgements

Sincere feelings and strongest kind words emanating from my heart go to my supervisor **Professor Xiangjian He** for his marvelous support, advice, help and practical suggestions throughout my PhD journey. I am greatly indebted to him for his excellent direction, patient instruction, steadfast encouragement and timely comments to this research throughout the past four years. I owe my research achievements to his excellent supervision. Without his support and supervision, I could not have come this far.

I would like to thank my co-supervisor, **Dr. Wenjing Jia** for her unflinching encouragement, support and her useful comments. Her valuable suggestions and guidance have been a great help and kept me moving ahead at a critical time. I would also like to thank my co-supervisor **Dr. Gengfa Fang** for his advice and assistance.

I am forever appreciate and thankful to my family for all of their love and encouragement. Special thanks to my father, Mr. Waleed Aldebei, and my mother, Mrs. Sameera Abu Alsamen, for their extraordinarily generous help during my study. Without their support, I would never have had a chance to study overseas.

I would also like to express my pure-hearted thanks to my wife, Eng. Alanoud Alnsour, for her love, understanding and patience. You have been a pillar of strength to me. I also thank my darling baby girl, Salma Aldebei, for entering into our lives and making it more fulfilling.

Finally, I would like to extend my thanks to all my colleagues, friends and the staff of the school, especially those people listed below for providing support and friendship that I needed.

- Professor Massimo Piccardi, Associate Professor Qiang Wu, Dr. Min Xu, Dr. Priydarsi Nanda, Shaukat Abedi, Sari Awwad, Ahmed Mian Jan, Minqi Li, Fairouz Hussein, Omar Alshaweesh and Muhammad Usman.

Contents

List of Figures

List of Tables

Abbreviations

AA	**A**uthorship **A**ttribution
DT	**D**ecision **T**ree
EM	**E**xpectation **M**aximisation
GMM	**G**aussian **M**ixture **M**odel
HMM	**H**idden **M**arkov **M**odel
IID	**I**ndependent and **I**dentically **D**istributed
IR	**I**nformation **R**etrieval
K-NN	**K**-**N**earest **N**eighbors
MAP	**M**aximum **A** **P**osterior
MLE	**M**aximum **L**ikelihood **E**stimation
NN	**N**eural **N**etworks
NLP	**N**atural **L**anguage **P**rocessing
NUANCE	**N**-gram **U**nsupervised **A**utomated **N**atural **C**luster **E**nsemble
PIP	**P**robability **I**ndication **P**rocedure
POS	**P**art-**O**f-**S**peech
SUDMAD	**S**equential and **U**nsupervised **D**ecomposition of a **M**ulti-**A**uthor **D**ocument
SVM	**S**upport **V**ector **M**achine
TF-IDF	**T**erm **F**requency-**I**nverse **D**ocument **F**requency
WAN	**W**ord **A**djacency **N**etwork

Publications

These are the publications resulted from this thesis.

- Khaled Aldebei, Xiangjian He, and Jie Yang. Unsupervised Decomposition of a Multi-Author Document Based on Naive-Bayesian Model. *Association for Computational Linguistics, Volume 2: Short Papers*, page 501, 2015. (**CORE A*, CCF A & ERA A**)

- Khaled Aldebei, Xiangjian He, Wenjing Jia, and Jie Yang. Unsupervised Multi-Author Document Decomposition Based on Hidden Markov Model. In *ACL (1)*, 2016a. (**CORE A*, CCF A & ERA A**)

- Khaled Aldebei, Helia Farhood, Wenjing Jia, Priyadarsi Nanda, and Xiangjian He. Sequential and Unsupervised Document Authorial Clustering Based on Hidden Markov Model. In *Trustcom/BigDataSE/ICESS, 2017 IEEE*, pages 379–385. IEEE, 2017. (**CORE A & ERA A**)

- Khaled Aldebei, Xiangjian He, Wenjing Jia, and Weichang Yeh. SUDMAD: Sequential and Unsupervised Decomposition of a Multi-Author Document Based on a Hidden Markov Model. *Journal of the Association for Information Science and Technology*, 69(2):201–214, 2018. ISSN 2330-1643. doi: 10.1002/asi.23956. URL http://dx.doi.org/10.1002/asi.23956. (**ERA A* & CCF B**)

Dedicated to My Family

Chapter 1

Introduction

This thesis addresses the problem of decomposing sentences of a multi-author document into components according to their authorship. Section 1.1 of this chapter presents the background for multi-author document decomposition process. The motivations for the work presented in this thesis and objectives are given in Section 1.2. The contributions and novelty of the work are explained in Section 1.3. Section 1.4 outlines the structure of the remainder of this thesis, followed by a summary of this chapter in Section 1.5.

1.1 Background

Research interest in document decomposition has increased as a result of the large growth rate of online documents that need to be analysed. Typically, document decomposition is a process of segmenting a document into components according to a specific criterion. Traditional studies on document decomposing, as shown in Brants et al. (2002), Hennig and Labor (2009) and Mota et al. (2016), focus on dividing a document into components based on topic, so that all texts in a component are relevant to only one topic. Furthermore, some other studies, as shown in Cesarini et al. (1999) and Duygulu and Atalay (2002), aim to decompose a document based on a regular layout using a specific type of document that contains tabular structures (i.e., invoice documents). Those works were done based on Natural Language Processing (NLP) techniques and different machine learning schemas.

Nowadays, with the evolution of online communication facilities, the cooperation of authors to produce a document becomes much easier. Therefore, it is not surprising that the amount of multi-author documents has drastically increased. Multi-author documents can be found in Web pages, books, academic papers and blog posts. Interestingly, although numerous approaches have been presented to handle various problems related to multi-author documents, very few of these approaches are based on decomposing a multi-author document according to authorship. Formally, authorship-based multi-author document decomposition is a process of segmenting sentences of a multi-author document into components according to their authorship, so that all sentences in any component are written by only one author. The main assumption made is that each sentence in the document is written by only one author. The process should be applied when no training data are available at all. An illustration of the process of decomposing a document written by N authors according to authorship can be seen in Figure 1.1.

Some researchers, such as Koppel et al. (2011a) and Daks and Clark (2016), have focused on authorship-based document clustering problem where the task is to group documents written by the same author in one cluster. Practically, this problem is quite different from, and easier than, the authorship-based multi-author document decomposition problem because in the authorship-based document clustering problem all sentences of each document are written by only a single author, and so an author's writing style of the document can be easily observed and distinguished from the other authors' writing styles of other documents. However, in the authorship-based multi-author document decomposition problem, because sentences of a document are irregularly written by more than one author, the process of differentiating writing styles among authors in the document is intrinsically hard. Note that in some works, such as Graham et al. (2005), the authors have addressed an easier version of the authorship-based multi-author document decomposition when it is assumed that each paragraph in the document is written by one single author.

Authorship-based multi-author document decomposition process has many advantages that make it attractive and essential for researchers in recent years. For example, it can be useful in defining the contributions of authors in a multi-author document, such as academic papers and theses. Furthermore, the process can be helpful in identifying the

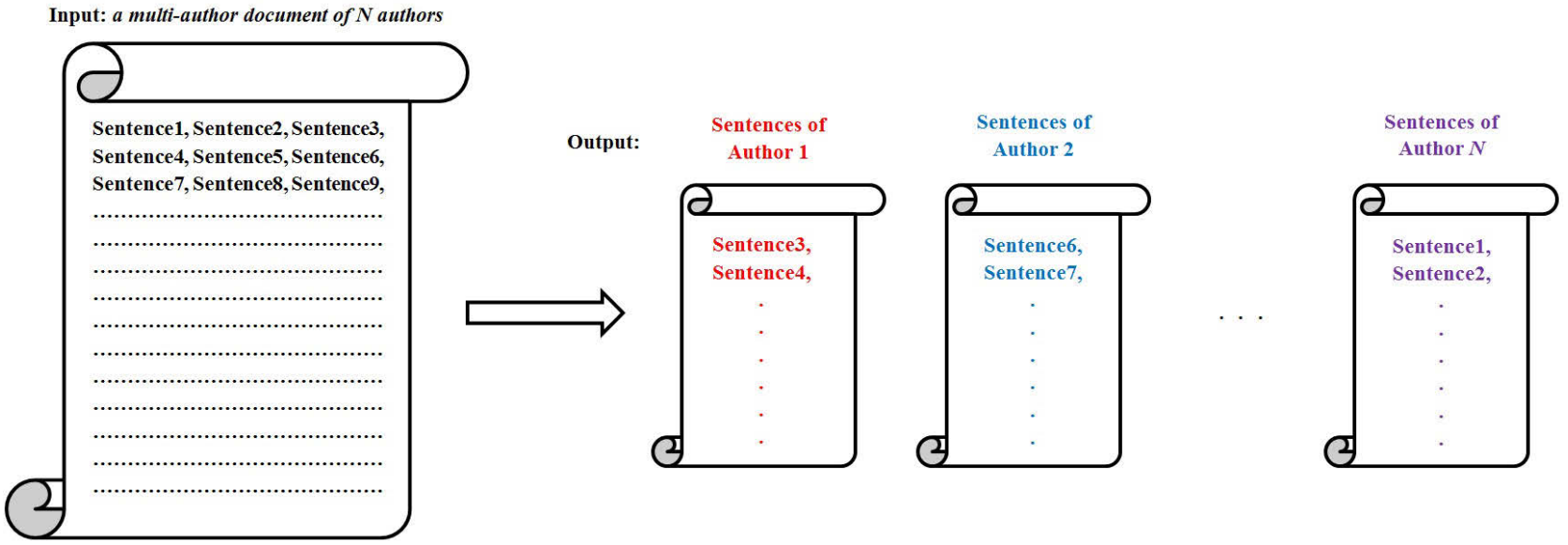

FIGURE 1.1: An illustration of the decomposing process of a document written by N authors.

true author of a piece of text (such as a ransom note) that has a doubt on authority (i.e., Civil Law problems) and so it may save lives or catch the offender. The process is also beneficial for forensic analysis problems, such as authorship attribution (Stamatatos, 2009a), where the objective is to determine the authorship of a disputed document using a set of training documents with known authorships. Another advantage of the multi-author document decomposition is on intrinsic plagiarism detection (Zu Eissen and Stein, 2006), where the task is to retrieve all plagiarized sentences from a document when no external sources are available.

Despite all of the aforementioned advantages of the authorship-based multi-author document decomposition process, its few existing studies suffer from considerable limitations, such as being applicable only to specific types of documents (i.e., Hebrew language documents), being restricted to a document of two authors only, being useful only when topics among authors are distinguishable, requiring a set of labeled sentences for training and being less accurate.

This thesis intends to address the aforementioned limitations and proposes approaches that can be efficiently applied for authorship-based multi-author document decomposition. These approaches are completely unsupervised and do not require the availability of any information of authors or document's context. They are efficacious even when the topics in the document are not detectable among authors. They are also language-independent approaches and can be used in a document written by any number of authors. Although there is a current research direction to overcome these limitation, still there are rooms for improvements.

1.2 Motivations and Objectives

Motivations

The quality of an authorship-based multi-author document decomposition system is defined by its effectiveness and adaptability. The effectiveness of a system is evaluated via its ability to correctly segment sentences of a multi-author document into components based on their authorship. The adaptability, however, is measured by the ability of

segmenting sentences of different types of document with less restrictive assumptions. Five main challenges need to be overcome through the development of authorship-based multi-author document decomposition systems, which have **motivated** this Ph.D research. These challenges are detailed as follows.

The first challenge is that a system for decomposing a multi-author document according to authorship must be used in a document where the topics among authors are not differentiated. Having more than one topic for each author or sharing topics among authors in the document makes the decomposing process harder. Several approaches on document decomposition rely on topics to differentiate the writing styles among authors. They assume that each author has different topics from other authors. Decomposing a multi-author document with single topic is also challenging because all sentences in the document represent only one topic, and so distinguishing the writing styles among authors is more difficult. As a benchmark authorship analysis dataset, Becker-Posner blogs dataset is a typical example of multi-topic and single-topic documents where the topics among authors are totally non distinguishable.

The second challenge is the unavailability of labeled training datasets. Decomposing sentences of a multi-author document into authorial components with an unsupervised learning scenarios, where no labeled data are available, is much more difficult than in supervised learning scenarios. That is, a system that can decompose a document into authorial components using only the sentences of the document with no any other information about the authors or the document should be developed. In fact, unsupervised learning for this system is desired because collecting training data of authors contributed to the document is very hard. For some cases, no training data of authors are available at all.

The third challenge is that the authorship-based multi-author document decomposition must be used in a document of any number of authors. It is very important for the authorship-based document decomposition systems that no limitation on the number of authors in the document is implied. The authors of Koppel et al. (2011a) have proposed an approach for decomposing a document into components based on authorship. However, their approach is only applicable for documents formed by two authors.

The fourth challenge is the ability to decompose a document into authorial components with high efficiency. It is very fundamental for document decomposition systems to achieve high accuracy in grouping sentences written by the same author in one component as possible. A significant amount of work has been carried out to address this challenge. Several machine learning and distance measurement techniques, including Support Vector Machine (SVM) Akiva and Koppel (2013), spectral clustering Koppel et al. (2011a) and cosine similarity measurement Akiva and Koppel (2012), have been employed to strengthen the decomposition process. However, their accuracies still need to be improved.

The fifth challenge is the difficulty of providing an authorship-based multi-author document decomposition system that can be used in any language document. Achieving that requires using a language-independent feature set to represent a document of any language. Some approaches, such as Koppel et al. (2011a), have used a customised feature sets that can only be applicable to specific types of documents (i.e., Hebrew language documents).

Objectives

The overall aim of this thesis is to develop novel authorship-based multi-author document decomposition approaches that are completely unsupervised and do not need the availability of any information of authors or document's context. The approaches should also be effective even when the topics in the document are not distinguishable among authors and when the number of authors is more than two. The specific objectives of this thesis are as follows.

1. We will propose multi-author document decomposition based on authorship approach that does not require any information about the document's context or authors' writing profiles. The approach should be able to handle documents of any types and any topics written by any number of authors.

2. We will develop a refinement procedure to improve the performance of the authorship-based multi-author document decomposition approach.

3. We will propose a procedure for creating an accurate labeled training dataset used to train a more powerful classifier in order to achieve better segmentation results.

4. We will propose a novel framework that uses the useful sequential correlations among consecutive sentences to group a sentences written by the same author.

1.3 Thesis Contributions

The main contributions of this thesis are summarised as follows:

- In Chapter 3 of this thesis, we develop a new unsupervised approach for segmenting a multi-author document into authorial components by exploiting the difference in the posterior probability of the Naive-Bayesian model to increase the efficiency of the clustering assignment and classification process. The proposed approach has the following properties.
  - It effectively selects the most discriminative data vectors from each cluster component based on the difference of the posterior probability and uses them to train a classifier.
  - It selects trusted sentences from a document and involves them to reclassify all sentences in the document.
- To enhance the performance of the approach presented in Chapter 3, a general unsupervised, two-level hierarchical learning framework for decomposing a document based on authorship is developed in Chapter 4. In this two-level learning framework, the purpose of the first level of learning is to generate a discriminative training dataset from unlabeled input data using a probability-based segment elicitation procedure, and use it to train the first-stage classifier. The results of the first-stage classifier are utilized to create a new but more accurate training dataset, which is then used for training the second-stage classifier in order to achieve better purity results.
- In Chapters 3 and 4, we assume that the sentences of a document are independent and identically distributed (iid), i.e., no consideration has been given to the

contextual information between the sentences. However, in some cases, the iid assumption is deemed as a poor one. Therefore, in Chapter 5, we utilize the sequential patterns hidden among document elements (i.e., sentences) when determining their authorships. The proposed approach has the following properties.

  - It captures the dependencies between consecutive elements in a document to identify different authorial components and constructs a Hidden Markov Model (HMM) for classification.
  - It initialises the HMM parameters using an unsupervised learning method.
  - Different from the approaches presented in Chapters 3 and 4, this unsupervised approach no longer relies on any predetermined threshold for document decomposition.

- In Chapter 6, we further extend our approach in Chapter 5 and propose a two-stage HMM model in order to utilize the sequential patterns among sentences more comprehensively. The proposed approach has the following properties.

  - It creates a new labeled training dataset to learn a more accurate HMM and further boost the performance of this approach.
  - It utilizes the contextual relationships among sentences in order to refine the classification results.

1.4 Thesis Structure

The remainder of this thesis is organised as follows. **Chapter** 2 reviews previous research work in the area of authorship analysis, stylometric features, feature representations and authorship analysis techniques. Furthermore, it introduces the Naive Bayesian method, Hidden Markov Model (HMM) and Gaussian Mixture Models (GMMs) and shows some of their applications in authorship analysis. **Chapter** 3 presents a new unsupervised method for decomposing a multi-author document into authorial components. A new procedure called "Segment Elicitation" for selecting best segments is presented. Furthermore, this chapter proposes a "Probability Indication Procedure" to increase the purity

results using five criteria. The two procedures (i.e., Segment Elicitation and Probability Indication) are based on exploiting the difference in the posterior probability of the Naive-Bayesian model. **Chapter** 4 proposes a general unsupervised, two-level hierarchical learning framework for segmenting a document into distinct authorial components. In this framework, results of the first level learning are utilized to generate a new but more accurate training dataset, which is then used for training the second level learning in order to achieve better purity results. The framework is evaluated on challenging benchmarks, such as Single-Topic Becker-Posner Blogs. A scientific paper is also used to show the application of the proposed approach to the authorship decomposition on an authentic document. The experimental results show that the approach is capable of achieving remarkable multi-author document decomposition results. The approaches presented in Chapters 3 and 4 assume that the sentences in a document are independent and identically distributed (iid) and no relation among sentences in the document. **Chapter** 5 presents a new unsupervised approach for segmenting a multi-author document into authorial components. The proposed approach uses the sequential patterns hidden among document elements when determining their authorships. For this purpose, a Hidden Markov Model (HMM) is adopted and a sequential probabilistic model is constructed to capture the dependencies of sequential sentences and their authorships. This chapter also proposes an unsupervised learning method for initializing the HMM parameters. Furthermore, the chapter presents an application of the proposed approach on Authorship Attribution (AA). **Chapter** 6 proposed an algorithm for Sequential and Unsupervised Decomposition of a Multi-Author Document (SUDMAD) through the construction of a Hidden Markov Model reflecting authors' writing styles. Within this algorithm, SequentialUD and a refined SequentialUD approaches for multi-author document decomposition are presented. Many experiments are included in this chapter to demonstrate the superior performance of the proposed approach. Finally, **Chapter** 7 summarises the contributions of this thesis.

1.5 Summary

This chapter has provided the reader with a background of this thesis. It includes an overview of a document decomposition process and an introduction to authorship-based multi-author document decomposition task. It has been shown that this task plays an increasingly important role in many applications and has great significance on security and forensic investigation. Despite its value, there are very few works reported. Existing works have limitations on topic, specific languages, styles of writing, or requiring the availability of the profiles of authors. These limitations are the motivations for this thesis research. Four different approaches of the authorship-based multi-author document decomposition will be presented in this thesis. The major contributions of the approaches are briefly listed in this chapter. An overview of the structure of this thesis is also presented at the end of this chapter.

Chapter 2

Background and Related Work

In this chapter, background information is given to introduce the reader to the works that have been achieved in the following chapters of this thesis. The outline of this chapter is as follows. Section 2.1 introduces the authorship analysis and reviews some of existing authorship analysis methods. Section 2.2 presents a Naive Bayes model and its applications in authorship analysis. Section 2.3 describes a well-known sequential model, Hidden Markov Model (HMM), and provides some previous HMM-based approaches on document analysis. Section 2.4 briefly discusses a classical Gaussian Mixture Model (GMM) and related Expectation Maximisation (EM) algorithm, and presents some GMM-based approaches on document analysis. Finally, a summary to the chapter is given in Section 2.5.

2.1 Authorship Analysis

Authorship Analysis is the task of analysing the characteristics of documents in order to draw conclusions on its authorships. Currently, authorship analysis has become more popular in determining and analysing the features of authorships of documents because of the huge number of documents on the Web and the needs for techniques to analyse their authorships. Furthermore, abusing and misusing the Internet makes the existence of effective authorship analysis techniques highly demanded. The documents include, for example, e-mail messages, scientific papers, blogs and online forums.

Authorship analysis is considered as a relatively old task. The first endeavor was on the 19^{th} century, when Mendenhall (1887) used word-length distribution statistics to identify the writing styles of Bacon, Marlowe and Shakespeare. In the first half of the 20^{th} century, some works have also been done on authorship analysis, such as the works of Zipf (1932), Yule (1939) and Backoff et al. (1944). These works have exploited vocabulary richness and sentence length to capture the authors' writing styles of documents. Another detailed work on authorship analysis was by Mosteller and Wallace (1964). The work addressed an authorship identification problem and attempted to solve the author debates over the *Federalist Papers* [1].

In recent time, the development of some scientific areas, such as Information Retrieval (IR), machine learning techniques and Natural Language Processing (NLP), has a strong impact on the development of authorship analysis field as follows.

- *Information Retrieval.* The researches on information retrieval propose efficacious methods for representing a large amount of text.
- *Machine Learning.* The researches on machine learning provide powerful techniques to classify text data more accurately. Furthermore, the techniques have an ability to handle multidimensional data.
- *Natural Language Processing* (NLP). The researches on natural language processing provide different tools for analysing text data in many different patterns.

Due to the impact of scientific areas, as described above, numerous approaches of authorship analysis have been developed. The approaches are adapted to handle different types of documents, such as text in literature (Burrows et al., 2002; Hoover, 2004), e-mails (De Vel et al., 2001; Estival et al., 2007; Iqbal et al., 2010), Web forum messages (Abbasi and Chen, 2005; Abbasi et al., 2008; Solorio et al., 2011), blogs (Koppel et al., 2006, 2011b; Akiva and Koppel, 2013), chat messages (Kucukyilmaz et al., 2006; Layton et al., 2010; Iqbal et al., 2013) and programming codes (Krsul and Spafford, 1997; Burrows et al., 2014; Alazab, 2015).

[1] *Federalist Papers* are a collection of 85 papers in favor of the ratification of the proposed United State Constitution.

Nowadays, authorship analysis has been applied in many diverse fields with great success (Stamatatos, 2009a). The fields include, for example, civil law (i.e., determining real author characteristics of a piece of text which has a doubt on authority) (Ginsburg, 2002; Ariani et al., 2014), forensic analysis (i.e., determining a real author of a disputed text given a set of candidate authors) (Grant, 2007; Iqbal et al., 2013), criminal law (i.e., determining authors of harassing messages and verifying the authenticity of suicide notes) (Zheng et al., 2003, 2006), computer forensic (i.e., determining authors of unclassified source code of malicious software) (Kothari et al., 2007; Bandara and Wijayarathna, 2013), plagiarism detection (i.e., finding similarities between two documents to detect a plagiarised text) (Potthast et al., 2010; Mirza and Joy, 2015) and marketing (i.e., determining what types of people like or dislike a specific product) (Ikeda et al., 2013; Jiang et al., 2015).

Due to the geographically unbounded nature of the Internet and the multilingual functionality of many online forums, different authorship analysis approaches have been developed to handle text of different languages. For example, the authors of Abbasi and Chen (2005) have proposed an approach for authorship identification for Arabic web forum messages. In Yu (2012), the authors have used function words for authorship attribution in Modern Chinese language documents. Furthermore, the work of Mikros and Argiri (2007) has investigated an authorship attribution task using Modern Greek language documents. However, some works, such as Peng et al. (2003) and Keselj et al. (2003), have developed language independent approaches for authorship analysis that can be applied on documents of different languages (i.e., Greek, English and Chinese). The work of Juola (2004) has also provided an authorship analysis approach that can be used in documents of various languages (i.e., English, French, Latin, Dutch and Serbian-Slavonic).

Most researches break down the authorship analysis task into five categories, i.e., authorship attribution, authorship verification, plagiarism detection, authorship profiling and authorship-based text decomposition. The five categories will be briefly introduced in the next subsection.

2.1.1 Authorship Analysis Categories

As illustrated before, the authorship analysis is the task of examining writing characteristics in order to make inferences about authorship. Five main categories are included in authorship analysis (Stamatatos, 2009a). Each category is associated with a certain task of authorship analysis. The five categories are shown as follows.

2.1.1.1 Authorship Attribution

Authorship Attribution (AA), or *Authorship Identification* as named in Zheng et al. (2003), is the process of identifying the real author of a disputed document given a set of labeled documents of candidate authors. The process involves analysing writing styles of documents knowingly written by the candidate authors to determine the authorship of the disputed document.

Several approaches have been shown to handle authorship attribution, such as the approaches of Eder (2013), Koppel et al. (2013), Brand et al. (2015) and Rocha et al. (2017). In the literature, two kinds of authorship attribution are presented. The first one is a *closed-set* authorship attribution (Diederich et al., 2003; Khonji et al., 2015), where the real author of a disputed document is one of the candidate authors. The second one is an *open-set* authorship attribution (Schaalje et al., 2011, 2013), where the real author of a disputed document may or may not belong to the candidate authors. The *open-set* authorship attribution is much more difficult than the *closed-set* authorship attribution especially when the size of the candidate author set is small (Koppel et al., 2011b). A special scenario of authorship attribution is called *Needle-in-a-Haystack* (Rappoport and Koppel, 2013; Nirkhi and Dharaskar, 2013). The scenario occurs when there are many thousands of candidate authors with a very limited writing samples for each candidate author.

2.1.1.2 Authorship Verification

Authorship Verification is the process of checking whether a disputed document (i.e., target document) was written or not by a certain author. In this category, there is only

one suspect rather than a set of candidate authors. The main question in authorship verification is *"Did the candidate author x write the document?"* In fact, authorship verification is a more realistic task than authorship attribution, since the set of candidate authors for a document is basically unknown. Forensic scientists not only want to recognise the real author given a small set of candidate authors, they also intend to make certain that the real author is not someone else not under investigation (Luyckx and Daelemans, 2008).

Several researchers have proposed and developed approaches for authorship verification. For example, Koppel and Schler (2004) have presented an authorship verification approach named "unmasking", which can only be successfully applied on long documents (i.e., documents of at least 500 words long). Furthermore, the authors of Chen et al. (2011) and Canales et al. (2011) have studied authorship verification on short email messages and exam documents, respectively.

2.1.1.3 Plagiarism Detection

Plagiarism Detection is the process of comparing two or more documents and finding degree of similarity among them. According to research, there are two basic types of plagiarism detection, i.e., *external plagiarism detection* and *intrinsic plagiarism detection*. The external plagiarism detection (Gupta et al., 2014; Vani and Gupta, 2014; Ravi et al., 2016) is mainly concerned with the comparison of contents of a suspect document against contents of a set of external documents (e.g., web pages, text books, etc.) in order to unveil portions that might be plagiarised. The external plagiarism detection is based on finding passages in the suspect document which were copied from other external documents. On the other side, the intrinsic plagiarism detection (Bensalem et al., 2014; Kuta and Kitowski, 2014; Wijaya and Wahono, 2015) is only concerned with detecting plagiarized portions from a suspect document without comparing it with any external documents. The intrinsic plagiarism detection is based on analysing the suspect document with respect to writing style changes.

2.1.1.4 Authorship Profiling

Authorship Profiling, or *Authorship Characterization*, is the process of inferring information -rather than identity- of an author of a disputed document. The authorship profiling is an important task in many real applications. For example, in forensic, the authorship profiling can help police to identify the characteristics of the criminal of the crime. Furthermore, the authorship profiling is useful in marketing when a large companies may be concerned to define what types of people like or dislike their products, based on analysing blogs and online product reviews.

Some particular author's information that was previously reported in a literature are age (Argamon et al., 2009; Villena Román and González Cristóbal, 2014), gender (Schler et al., 2006; Mechti et al., 2014), educational level (Corney et al., 2002; Estival et al., 2007), language background (Koppel et al., 2005; Estival et al., 2007), political orientation (Koppel et al., 2009a) and occupation (Pham et al., 2009).

2.1.1.5 Authorship-Based Text Decomposition

Authorship-Based Text Decomposition is the process of clustering texts into components according to authorship. Unlike the authorship attribution, the authorship-based text decomposition does not require a set of labeled documents to be employed as training data. Therefore, unsupervised learning models that are able to capture similarities or differences among authors' writing styles in texts are expected to be built.

Two main versions of the authorship-based text decomposition can be discerned, i.e., Authorship-Based Document Clustering and Authorship-Based Multi-Author Document Decomposition.

Authorship-Based Document Clustering

Authorship-based document clustering is the process of clustering a group of single-author documents into authorial components. Each component contains the documents that are written by the same author. The authorship-based document clustering is

strongly related to authorship verification (Koppel and Winter, 2014; Stamatatos et al., 2014). Many applications can take advantages of this process. For example, suppose that there are a group of single-author documents (e.g., novels, blogs, papers, product reviews) with anonymous authors. Then, by applying an effective authorship-based document clustering technique, we can extract useful conclusions, such as that a group of anonymous novels is written by a single author or a group of blogs, which have same alias, is in fact written by different authors.

Very few works have been reported in the literature on the authorship-based document clustering. For example, the work of Koppel et al. (2011a) proposed an unsupervised approach for clustering chapters of Bible books written by two different authors. In Daks and Clark (2016), the authors developed a document clustering approach using part-of-speech (POS) feature sets.

Authorship-Based Multi-Author Document Decomposition

Author-based multi-author document decomposition is the process of decomposing a multi-author document into authorial components. Each component contains the sentences that are written by the same author. The author-based multi-author document decomposition is considered more difficult and challenging than the first version (i.e., authorship-based document clustering). That is because in the first version, since all sentences in each document are written by only one author, authors' writing styles of documents can be easily captured and differentiated. However, in the author-based multi-author document decomposition, since sentences of a document are written by multiple authors without a specific order, authors' writing styles implied in the document are hard to be captured and differentiated.

The author-based multi-author document decomposition has a great practical importance in forensic analysis, civil law, criminal law, plagiarism detection and intelligence issues. For example, it can be utilised to estimate the contribution of each author in a collaborative document (e.g., thesis, scientific paper). Furthermore, it can be used as an evidence to determine the real author of a piece of text (such as a ransom note) that has a doubt on authority and so it may save lines or catch the offender.

Despite the above-mentioned importance of the authorship-based multi-author document decomposition, only very few works are reported in the literature in this regard. In Graham et al. (2005), the authors proposed a supervised approach assuming that each paragraph was written by only one single author. They trained their approach using a set of labeled pairs of paragraphs. Each pair of paragraphs contains two paragraphs written by the same author or two different authors, while each paragraph is written by only one author. The authors of Koppel et al. (2011a) are the first researchers who implemented an unsupervised approach for decomposing a two-author document into two authorial components. In their approach, each paragraph is not necessarily written by a single author. However, the authors employed a feature set consisting of 1595 synonyms that are only applicable on particular types of documents such as Bible books written in Hebrew. Furthermore, the approach required specific tools for identifying synonyms in biblical books. The approach is also only useful for a document written by two authors. The work of Akiva and Koppel (2012) investigated the limitations in the approach of Koppel et al. (2011a) and presented an generic unsupervised approach. The approach utilised distance measurements to increase the precision and accuracy of clustering and classification phases, respectively. However, the resultant accuracy of the approach was not satisfactory. In Akiva and Koppel (2013), the authors further improved their original work presented in Akiva and Koppel (2012) and proposed an effective approach for the authorship-based multi-author document decomposition. They also utilised distance measurements to improve the efficiency of the proposed approach. However, The accuracy of their approach is highly dependent on the number of authors. When the number of authors increases, the accuracy of the approach drops significantly. For the same purpose, the author of Giannella (2015) presented a new approach named BayesAD, where the number of authors of the document can be either known or unknown. In his approach, a Bayesian segmentation algorithm is applied, which is followed by a segment clustering algorithm. The approach was tested on short documents (i.e., the number of sentences in a document is less than 500). However, the approach was tested by using only documents with a few transitions among authors. Furthermore, the performance of the approach is very sensitive to the setting of its parameters.

In this thesis, we address the authorship-based multi-author document decomposition

and develop new approaches that can effectively and efficiently decompose a multi-author document into authorial components.

Generally, the main task of the categories of authorship analysis mentioned before (i.e., authorship attribution, authorship verification, plagiarism detection, authorship profiling and authorship-base text decomposition) is to capture and define authors' writing styles and differentiate among them. Typically, the authors' writing styles are captured by employing an appropriate feature set that is used to vectorise text data by considering a certain representation form. A proper approach is then applied on the resulted feature vectors in order to identify the writing styles of authors. In the following three subsections, we will discuss these three main factors in authorship Analysis (i.e., stylometric features, feature representations and approaches for authorship analysis), respectively.

2.1.2 Stylometric Features

The main idea of capturing and discriminating the authorships of text data is by extracting the appropriate features from the text data which can differentiate the fingerprints among authors (Stamatatos, 2009a). The development in statistical and machine learning methods allows researchers to consider a wide variety of different types of features, and provides techniques that can effectively handle multidimensional and sparse data. Furthermore, the availability of Natural Language Processing (NLP) tools provides an ability to efficiently analyse text and produce new forms of measurements for representing writing styles.

In fact, the selection of an appropriate feature set, which can capture the writing styles of authors, is one of the important factor in authorship analysis because it may significantly affect the performance of authorship definition. In this subsection, some feature types that have been, or will be, utilised for authorship analysis are presented.

2.1.2.1 Syntactic Features

Syntactic features are deemed as one of the most important features used in authorship analysis. The implicit point of the syntactic features is that authors tend to unintentionally employ same syntactic patterns in their writings.

Usually, the extraction process of syntactic features requires robust and accurate NLP tools (i.e., parsers) to be available. The tools should be qualified for analysing a particular natural language with comparatively high performance. Therefore, the extraction process of syntactic features is highly language dependent.

Many works of authorship analysis have been done by utilising the syntactic features in representing documents for the purpose of identifying authors' writing styles of the documents. In Baayen et al. (1996), the authors were the first to employ syntactic feature measurements for representing documents and determininig authors' writing styles. They have used a full parse tree to describe two different aspects. The first aspect is what the syntactic class of each word is. The second aspect is how the words are combined to form phrases. For example, the following rule:

$$\text{A : PP} \longrightarrow \text{P : PREP} + \text{PC : NP}$$

means that an adverbial prepositional phrase is defined by a preposition followed by a noun phrase as a prepositional complement (Stamatatos, 2009a).

The authors of Stamatatos et al. (2000), Stamatatos et al. (2001), Gamon (2004) and Hirst and Feiguina (2007) have also exploited NLP parsers in order to produce syntactic patterns. They have utilised the frequencies of these patterns to define the writing styles of authors.

Examples of syntactic patterns that are exploited in authorship analysis include the following.

- Noun Phrase.
- Proper Noun Phrase.

- Determiner Phrase.
- Preposition Phrase.
- Adjective Phrase.
- Plural Nouns Phrase.
- Verb Phrase.

Some authors, such as those for Diederich et al. (2003), Zheng et al. (2006), Zhao and Zobel (2007) and Qian et al. (2014), have used a Part-Of-Speech (POS) tagging method to detect the actual tag of each word (i.e., verb, noun, adjective, etc.). The authors have exploited POS tag frequencies and POS tag n-gram frequencies to represent documents and capture author's writing styles.

Another interesting use of syntactic features in authorship analysis was proposed in the approach of Koppel and Schler (2003) where syntactic error information, such as sentence fragments and mismatched tense, have been detected using a commercial spell checker and utilised for representing document styles.

Recently, the approach of Daks and Clark (2016) has made use of syntactic structure for document clustering based on authorship. In that approach, POS n-grams have been utilised for identifying an individual writer.

2.1.2.2 Lexical Features

A simple way to represent a text is as a sequence of tokens grouped into sentences, with each token corresponding to a word, number, or punctuation mark. Some approaches of authorship analysis have employed measurements of these tokens and used it for representing the text and recognising authors' writing styles. Some examples of the token measurements are the mean number of words per sentence, the standard deviation of the number of words per sentence (i.e., sentence length variation) and the count of commas or colons per sentence. The extraction process of lexical features is language independent because such type of features can be extracted and used to represent the

text written in any language without requiring any extra tools or information of language structure.

Some works on authorship analysis, such as the works done in Luyckx and Daelemans (2008), Akiva and Koppel (2012) and Qian et al. (2014), have exploited *vocabulary richness* functions to measure the lexical diversity of the text. They have computed the ratio of the number of the unique tokens in the text to the total number of all tokens in the text. However, the number of tokens in the text strongly depends on the length of the text. Therefore, another set of lexical features that are not dependent on text length has been constructed. The set contains a group of function words (e.g., prepositions, pronouns, auxiliary verbs). The function words are topic independent and are able to capture writing styles of authors across different topics because they do not carry any semantic information but serve to express grammar relationships with other words. In literature, different sets of function words have been proposed and used for authorship analysis. For example, the works of Argamon et al. (2003), Abbasi and Chen (2005) and Argamon et al. (2007) have proposed sets of 150, 303 and 675 function words, respectively.

The vast majority of authorship analysis researches are based on using a set of bag-of-words for representing the text. The representation process starts by deeming the text as a set of unique words. Each word in the set has a frequency of occurrence disregard of contextual information. Then, most common words (i.e., words of highest frequencies) are selected and used for representing the text in order to define authors' writing styles (Burrows, 1987; Argamon and Levitan, 2005). Different bag-of-words sets have been used in authorship analysis for text representation. For example, the authors of Koppel et al. (2011a) have used a set of 223 lexical words to vectorise the text. Furthermore, the work of Akiva and Koppel (2012) has utilised a set of the most 500 words for representing a document in a multi-author document decomposition problem. Likewise, in Savoy (2013a), the authors have created a feature set containing the most 50 words occurred in their texts. On the other hand, several approaches of authorship analysis have created different feature sets containing all words that appear at least k times in their text. For example, a feature set of all words occurring at least two times and five times in a text has been employed in the approaches of Koppel et al. (2011a)

and Akiva and Koppel (2013), respectively. Furthermore, a feature set containing all words occurring in a text has been used in Savoy (2013b).

2.1.2.3 Application Specific Features

The variety of documents used in authorship analysis, such as E-mail messages and online documents, reveals a possibility to define new feature sets that are directly relevant to some specific documents. For instance, the approach of Koppel et al. (2011a) has created a new feature set able to distinguish writing styles of authors in Hebrew Bible books. The feature set includes 1595 synonyms written in Hebrew language. Furthermore, the authors of Roffo et al. (2013) have developed a new feature set able to better discriminate authors' writing style of online messages and chats. The feature set includes writing speed, mimicry and answering time. Other features, such as signature types, paragraph lengths, font-color count, font-size counts and the use of indentation, have also been used in researches of authorship analysis.

Some other approaches have developed a generalised feature set comprising a mixture of different types of features for describing stylometric styles (Chitrakar and Franke, 2014).

2.1.3 Feature Representation

The feature representation is the way of using the extracted stylometric features to form feature vectors, which are then utilised by one of machine learning techniques or statistical methods in order to achieve a certain task. A feature vector includes a set of elements, where each element of the feature vector is associated with a corresponding feature. Different feature representations have been involved for authorship analysis. The most common representation is a feature-frequency representation. In this representation, the value of each element of the feature vector represents the frequency of the corresponding feature in a text. That is, each feature vector forms a sequence of integer numbers, in which each number is equal to or more than zero. Many approaches, such as the approaches in Koppel et al. (2011a), Savoy (2013a) and Qian et al. (2014) have utilised feature-frequency representation to form feature vectors. Since this type

of feature representation depends on text length (i.e., when the text length increases, the frequencies of features increase), different methods have been used to normalise the resulted feature vectors. For example, a Frobenius norm (2-norm) is a common way used for normalising the numeric vectors. Furthermore, a Term Frequency-Inverse Document Frequency (TF-IDF) is one of the most important methods for normalising and weighting data, where each feature of the feature set is given a term-frequency that reflects the importance of the feature. In other words, a higher value is assigned to a feature if the feature occurs in a particular text and very seldom anywhere else, and a lower value is assigned to a feature if the feature occurs in each text.

Another way widely used for feature representation in authorship analysis is a feature-binary representation. In this representation, each element of the feature vector takes a value of 1 or 0, with 1 indicating the corresponding feature in a set appears in the text and 0 indicating not. That is, each feature vector forms a sequence of binary values. Examples of authorship analysis approaches that have employed the feature-binary representation to form feature vectors include Akiva and Koppel (2012) and Akiva and Koppel (2013).

In Segarra et al. (2014), the authors have adopted a different way to form feature vectors. They have used a normalised Word Adjacency Networks (WANs), which show a relation between every pair of function words in a document.

Literatures show that there is no best way for feature representation in authorship analysis, because there are many factors that affect the performance of each representation, such as the length of a text and the number of candidate authors. For example, in WANs, it requires that a text should be long and the number of candidate authors is small.

2.1.4 Approaches for Authorship Analysis

Over the last several years, a wide variety of approaches have been applied for authorship analysis. Some invariant approaches have been first proposed to study the authorships of text. The approaches are based on using some statistical measurements to differentiate writing styles among authors. These measurements include average word length

(Mendenhall, 1887; Mascol, 1888), average letter length (Brinegar, 1963) and average number of words in a sentence (Yule, 1946; Morton, 1965). Although this type of approaches has been adopted in many authorship analysis studies, it has not been proved stable (Sichel, 1986).

The non-stability of the invariant approaches has forced researchers to apply multivariate approaches for authorship analysis. In a lot of cases, a distance measure, such as Delta rule (Jockers and Witten, 2010; Savoy, 2013a), cosine similarity (Kjell et al., 1994; Koppel et al., 2011b) and Chi-square distance (Grieve, 2007; Luyckx and Daelemans, 2008; Savoy, 2013b), is defined and considered to distinguish writing styles among authors. For example, the most likely author of a document is the one that is corresponding to the smallest distance.

Recently, due to the vast improvements in machine learning and pattern recognition methods, many computer scientists have applied different machine learning and pattern recognition methods for authorship analysis task. Examples for such methods employed for this task are Neural Networks (NN) (Hoorn et al., 1999; Zheng et al., 2006), Naive-Bayesian (Clement and Sharp, 2003; Zhao and Zobel, 2005; Altheneyan and Menai, 2014), Support Vector Machine (SVM) (De Vel, 2000; De Vel et al., 2001; Akiva and Koppel, 2012, 2013), k-Nearest Neighbors (k-NN) (Abou-Assaleh et al., 2004; Halvani et al., 2013), decision tree (Diederich et al., 2003; Cheng et al., 2011), Bayesian regression (Madigan et al., 2005; Argamon et al., 2009) and random forest (Popescu and Grozea, 2012; Daks and Clark, 2016).

In authorship analysis tasks, data are usually either *independently and identically distributed* (iid) or *sequential* manner. Two events are said to be iid if the occurrence of the first event does not provide any information as to whether the second event occurs or not. For example, defining the authorship of one document does not help in defining the authorship of another document. However, in sequential data, it is assumed that the data form sequences. These sequences provide valuable sequential relations which can enhance the prediction accuracy of classifiers. For example, identifying the writer of one line is of great help for identifying the writer of the next line in handwriting document. Throughout the years, several models have been proposed to handle iid and sequential

data. The next two sections of this chapter (i.e., Section 2.2 and 2.3) present two different models for iid and sequential data, i.e., Naive Bayes and Hidden Markov Model (HMM), respectively. These two models are used in the next chapters to develop new approaches for the authorship-based multi-author document decomposition problem.

The majority of practical machine learning methods accepted for authorship analysis use *supervised learning algorithms* (see Figure 2.1). The idea of the supervised learning algorithms is that we have training data of input variables and an output variable and we use an algorithm to learn the mapping function from the input to the output. This learned function is then run to predict the output variable of new input variables. The approaches of Pearl and Steyvers (2012) and Hurtado et al. (2014), for example, have employed a set of labeled documents in order to define the writing styles of authors. On the other hand, some other machine learning methods accepted for authorship analysis use *unsupervised learning algorithms* (see Figure 2.2). In this case of learning, no desired outputs are available and the input data are only utilized in order to optimize a mapping function. The approach of Layton et al. (2013), for example, has proposed an unsupervised method to cluster a group of anonymous single-author documents according to authorship by applying an NUANCE (N-gram Unsupervised Automated Natural Cluster Ensemble). Some other machine learning methods, however, apply *semi-supervised learning algorithms* for authorship analysis. This type of learning stands between supervised and unsupervised learning and is operated when we have a large amount of unlabeled data but only a small number of labeled data. The approach of Qian et al. (2014), for example, has used a semi-supervised learning methods for authorship identification task.

As mentioned before, this thesis proposes unsupervised approaches for authorship-based multi-author document decomposition problem. One of the popular techniques applied for unsupervised learning is Gaussian Mixture Models (GMMs) clustering. In this thesis, the GMMs are considered in the proposed approaches and constructed to handle the unlabelled text data. Section 2.4 of this chapter gives a brief overview of the GMMs.

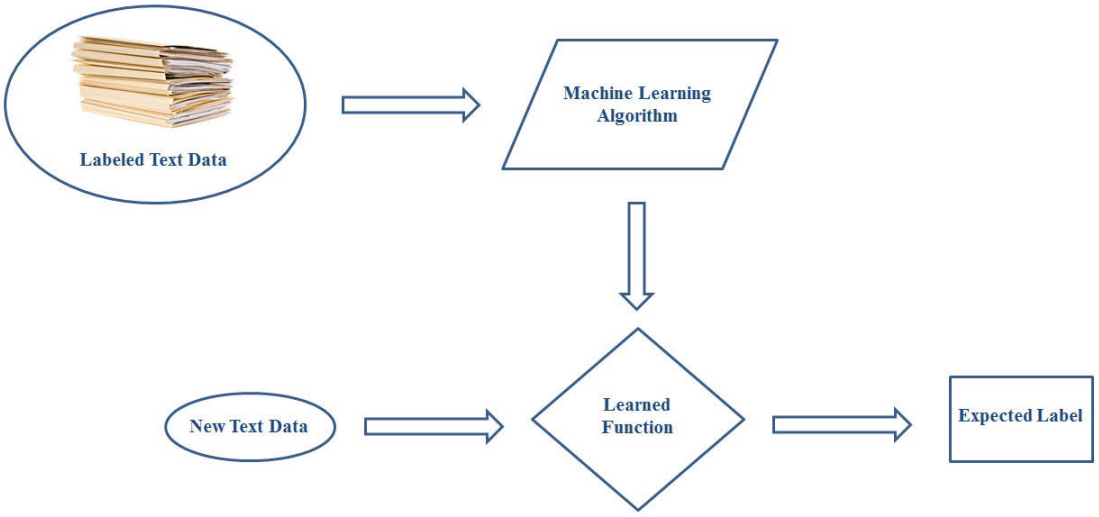

FIGURE 2.1: An illustration of the supervised learning of text data.

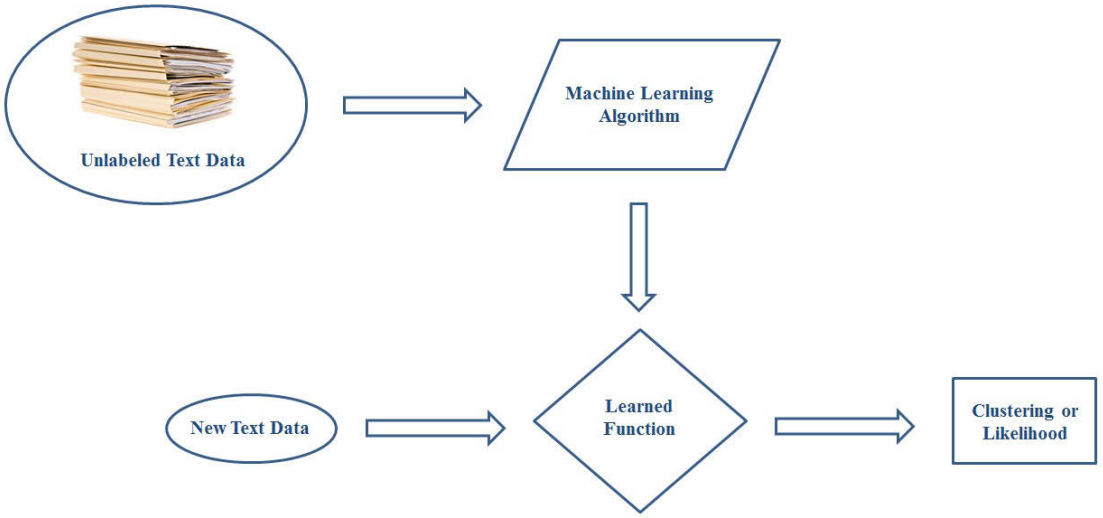

FIGURE 2.2: An illustration of the unsupervised learning of text data.

2.2 Naive Bayes

Naive Bayes model is one of the most well-known algorithms in classification and widely used in many applications. The Naive Bayes model is based on Bayes' theorem with the "naive" assumption of independence between every pair of features, i.e., it is assumed that the presence of a specific feature does not have any relation with the presence of any other features. In the following two subsections, the Bayes theorem and a Naive Bayesian classifier will be discussed, respectively. Furthermore, how Naive Bayesian can be applied in document analysis will be described in the last subsection of this section and supported with examples.

2.2.1 Bayes' Theorem

Bayes' theorem, which is named after Reverend Thomas Bayes (1701-1761), is used to estimate the probability of a specific event, based on prior knowledge of conditions that could be related to the event. For example, if lung cancer is related to smoking, then, by using Bayes' theorem, a person's smoking status (i.e., smoker or nonsmoker) can be used to more accurately determine the probability that the person has lung cancer or not, compared to the determination of the probability of having lung cancer without knowledge of the person's smoking status.

Let $\mathbf{A} = \{a^1, a^2, \ldots, a^D\}$ be a data sample. The data sample has values of D features. In Bayesian terms, $\mathbf{A}$ is considered as "evidence". Let B be a hypothesis, such that the data sample $\mathbf{A}$ is a member of class C. For classification purpose, the goal is to determine the probability that the hypothesis B is true given the "evidence" $\mathbf{A}$, $p(B|\mathbf{A})$. In other words, the probability that the data sample $\mathbf{A}$ is a member of class C, given the data sample $\mathbf{A}$, is what we are looking for.

To explain, let us go back to the example of lung cancer. Suppose that the data sample, $\mathbf{A}$, represents a person and it has one feature (i.e., $D = 1$). The feature is the person's smoking status (i.e., smoker or nonsmoker). Let us suppose that the value of the feature is "smoker". The hypothesis B represents that the person gets a lung cancer. Then

$p(B|\mathbf{A})$ is the probability that person **A** gets a lung cancer given that we know the person's smoking status, which is "smoker".

In contrast, $p(B)$ is the prior probability of B. It represents the probability that any person gets a Lung cancer regardless of person's smoking status, or any other features. Similarly, $p(\mathbf{A}|B)$ is the probability that a person **A** is smoker given that the person gets Lung cancer. $p(\mathbf{A})$ is the prior probability of **A**. It represents the probability that a person **A** is smoker.

According to the Bayes theorem, the probability that is requested to be computed (i.e., $p(B|\mathbf{A})$) is calculated as shown in Eqs. 2.1 and 2.2.

$$p(B|\mathbf{A}) = \frac{p(\mathbf{A}|B)p(B)}{p(\mathbf{A})}, \tag{2.1}$$

$$p(\mathbf{A}) = \sum_{all\ status\ of\ B} p(\mathbf{A}|B)p(B), \tag{2.2}$$

where $p(\mathbf{A}) \neq 0$ and

- $p(B|\mathbf{A})$ is the posterior probability of B given **A**.
- $p(B)$ is the prior probability of B.
- $p(\mathbf{A}|B)$ is the likelihood (i.e., the probability of **A** given B).
- $p(\mathbf{A})$ is the prior probability of **A**.

2.2.2 Naive Bayesian Classifier

Typically, a classifier is a function that maps input feature vectors $X = \{x_1, x_2, \ldots, x_N\}$, where N is the number of the vectors, to output class labels $Y = \{y_1, y_2, \ldots, y_M\}$, where M is the number of the classes. It is assumed that each input feature vector $x_i, i \in \{1, 2, \ldots, N\}$, represents $\mathbb{R}^D$ or $\{0, 1\}^D$. The first representation (i.e., $\mathbb{R}^D$) means that each feature vector is a group of D real numbers. The second representation (i.e.,

$\{0,1\}^D$), however, means that each feature vector is a group of D binary bits. Usually, the main objective is to learn the function of the classifier using a labeled training dataset (i.e., labeled vectors). Notice that this is an example of supervised learning (Kotsiantis et al., 2007).

In this section, we focus on a probabilistic classifier, which returns $p(y_j|x_i)$, where $j \in \{1,2,\ldots,M\}$ and $i \in \{1,2,\ldots,N\}$. The $p(y_j|x_i)$ is called a posterior probability and it is the probability of assigning y_j into x_i conditioned on x_i. Accordingly, the feature vector, x_i, is labeled with the class that achieves highest posterior probability, conditioned on x_i. In other words, the feature vector, x_i, is predicted to have label y_j if and only if

$$p(y_j|x_i) > p(y_{j'}|x_i) \quad \text{for} \;\; 1 \leqslant j' \leqslant M, \;\; j' \neq j. \tag{2.3}$$

Therefore, in order to assign a label to the feature vector (i.e., x_i), we find the class y_j that maximizes the posterior probability $p(y_j|x_i)$. This process is called maximum a posterior (MAP) hypothesis.

The posterior probability, $p(y_j|x_i)$, can be estimated using Bayes theorem as follows.

$$p(y_j|x_i) = \frac{p(x_i|y_j)p(y_j)}{p(x_i)}. \tag{2.4}$$

Since the prior probability of the feature vector (i.e., $p(x_i)$) is constant for all classes, the posterior probability is proportional to the likelihood of the training feature vectors times the prior probability of a class, as shown in Eq. 2.5.

$$p(y_j|x_i) \propto p(x_i|y_j)p(y_j). \tag{2.5}$$

That is, only $p(x_i|y_j)p(y_j)$ needs to be maximized, where $p(y_j)$ is the class prior probability and $p(x_i|y_j)$ is the likelihood probability. For simplicity, we substitute each feature vector x_i, $i \in \{1,2,\ldots,N\}$ with x as shown in Eq. 2.6.

$$p(y_j|x) \propto p(x|y_j)p(y_j). \tag{2.6}$$

2.2.2.1 Class Prior Probability

The prior probability of class y_j, i.e., $p(y_j)$, where $j \in \{1, 2, \ldots, M\}$, is estimated by considering y_j as a multinomial random variable as shown in Eq. 2.7.

$$p(y_j = r|\pi) = \pi_r, \tag{2.7}$$

where π is a M-dimensional vector of class probabilities.

By assuming that there are N' training labeled vectors, $\{(x_1, y_1), (x_2, y_2), \ldots, (x_{N'}, y_{N'})\}$, used to train the classifier, the maximum likelihood of π is estimated as follows.

$$\pi_r = \frac{\sum_{n=1}^{N'} I(y_n = r)}{N'} = \frac{{N'}_r}{N'}, \tag{2.8}$$

where $I(y_n = r)$ is an indicator function equal to 1 if $y_n = r$ and 0 otherwise, and ${N'}_r$ represents the number of training vectors labeled with r.

Note that if a class label does not occur in the training dataset (i.e., ${N'}_r = 0$), then the class prior probability, π_r, is equal to zero. Unfortunately, a zero estimate probability can cause significant problem when we classify a new input that has not been detected in the training dataset. Therefore, in order to solve this problem, Laplace smoothing (Lidstone, 1920; Johnson, 1932; Manning et al., 2008a) is used in order to prevent zero probability for class prior probability. The class probability with Laplace smoothing is given in Eq. 2.9.

$$\pi_r = \frac{{N'}_r + \alpha_r}{N' + \alpha}, \tag{2.9}$$

where α_r is some constant and $\alpha = \sum_r \alpha_r$.

A special case of Laplace smoothing, which is widely used in document analysis, is *add one* smoothing (Martin and Jurafsky, 2000; Hazimeh and Zhai, 2015). In this type of smoothing, a value of 1 is assigned to α_r as shown in Eq. 2.10.

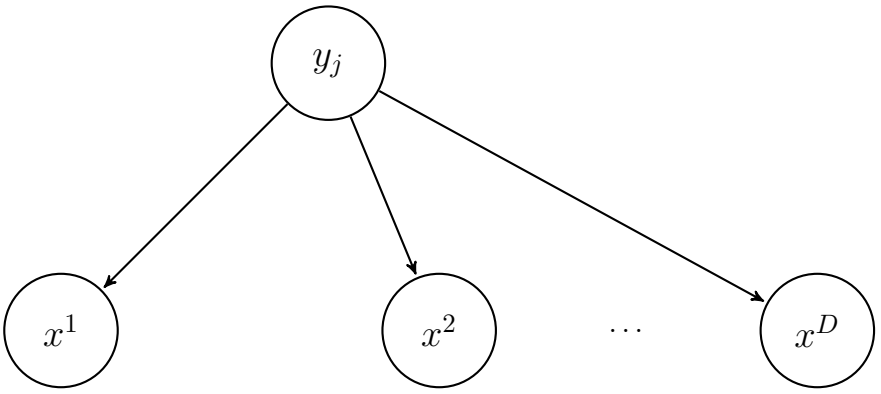

FIGURE 2.3: The conditional independent assumption of features in vector $x = \{x^1, x^2, \cdots, x^D\}$ given the class y_j.

$$\pi_r = \frac{N'_r + 1}{N' + M}, \tag{2.10}$$

where M is the number of classes.

For some cases, if the prior probabilities of all classes (i.e., $p(y_j)$, $j = 1, 2, \ldots, M$) can not be estimated, then it is assumed that all classes have the same prior probability, that is, $p(y_1) = p(y_2) = \cdots = p(y_M) = 1/M$.

2.2.2.2 Likelihood Probability

In order to estimate the likelihood probability, $p(x|y_j)$, the "naive" assumption (i.e., every pair of features are conditionally independent given the class) is made. Suppose that each input feature vector contains D features, i.e., $x = \{x^1, x^2, \ldots, x^D\}$. The independent assumption among features is illustrated in Figure 2.3.

Therefore, the likelihood probability, $p(x|y_j)$, is estimated as shown in Eq. 2.11.

$$p(x|y_j) = \prod_{d=1}^{D} p(x^d|y_j), \tag{2.11}$$

where x^d is the d^{th} feature of vector x.

The value of $p(x^d|y_j)$ can be easily estimated using the training dataset (i.e., N' labeled vectors). In normal cases, the value of $p(x^d|y_j)$ is the number of times that the feature

x^d occurs in training vectors labeled as y_j, divided by the number of training vectors labeled as y_j.

In case that a vector x contains discrete features, where each feature is a boolean value, i.e., $x^d \in \{0, 1\}$, then a Bernoulli distribution can be used to estimate $p(x^d|y_j)$, as follows.

$$p(x^d|y_j, \theta_{dj}) = \theta_{dj}^{x^d}(1 - \theta_{dj})^{1-x^d}, \tag{2.12}$$

where θ_{dj} is a probability of class y_j to generate the feature x^d.

The value of θ_{dj} is the number of times that the feature x^d occurs in training vectors labeled as y_j, divided by the number of training vectors labeled as y_j.

However, if the features of vector x are continuous-values, then it is commonly assumed that the values associated with each class are distributed according to a Gaussian distribution with a mean μ and a standard deviation σ. The $p(x^d|y_j)$ of this case is estimated as shown in Eq. 2.13.

$$p(x^d|y_j) = g(x^d, \mu_j, \sigma_j), \tag{2.13}$$

The $g(x^d, \mu_j, \sigma_j)$ is a Gaussian distribution defined as follows.

$$g(x^d, \mu_j, \sigma_j) = \frac{1}{\sqrt{2\pi\sigma_j}} \exp - \frac{(x^d - \mu_j)^2}{2\sigma_j^2}, \tag{2.14}$$

where μ_j and σ_j are the mean and the standard deviation of values in x^d of training vectors labeled as y_j, respectively.

After all, to predict the class label of feature vector x, we find the value of $p(x|y_j)p(y_j)$ for each class (i.e., $j = \{1, 2, \ldots, M\}$). Then, we select the class that maximizes the value of $p(x|y_j)p(y_j)$ and assign it to the feature vector x, as indicated in Eq. 2.15.

$$\hat{y} = \underset{y_j}{\operatorname{argmax}}\, p(y_j) \prod_{d=1}^{D} p(x^d|y_j), \qquad j = 1, 2, \cdots, M, \tag{2.15}$$

where $\hat{y}$ is the predicted class label.

Generally, the Naive Bayesian classifier is an important technique in many classification problems. The importance of the classifier is due to its remarkable properties, such as its simple representation, its speed and its good performance with a small training dataset. However, despite these advantages, there are some disadvantages associated with this classifier. One of these disadvantages is its strong assumption of independency between every pair of features. This "naive" assumption makes the method not applicable for many real-world tasks where the dependence among features is determined.

2.2.3 Naive Bayesian in Document Analysis

Document Analysis is the process of analysing a document in order to study and determine the characteristics of the document. Due to the huge number of documents on the World Wide Web, and the need for techniques for analysing these documents, document analysis is becoming very important in recent years in many fields of information retrieval and linguistic analysing. The documents include, for example, saved Web pages, email messages, scientific papers, reports, etc. According to the literature, document analysis involves a variety of tasks. Document Classification is one of the tasks that has received considerable research attention in recent years. The Document Classification is the task of assigning documents, or parts of a document, into predefined categories or classes. One interesting example is the process of classifying given documents into two categories (i.e., classes), "spam" or "non-spam". There are many other examples of document classification, such as Web pages classification, topic categorization, authorship attribution and polarity detection.

Numerious approaches have widely been employed in the document classification, including Support Vector Machine (SVM) (Joachims, 1998; Tong and Koller, 2001; DOrazio et al., 2014), Decision Tree (DT) (Li and Jain, 1998; Johnson et al., 2003; Farid et al., 2014), Naive Bayes classification (McCallum et al., 1998; Pop, 2006; Ting et al., 2011), K-Nearest Neighbors (KNN) classification (Han et al., 2001; Zhang and Zhou, 2005; Jiang et al., 2012), Neural Network (NN) (Farkas, 1993; Manevitz and Yousef, 2007; Moraes et al., 2013) and maximum entropy technique (Nigam et al., 1999; Zhu et al.,

2005; El-Halees, 2015). Among these approaches, Naive Bayesian method is considered as one of the most common techniques widely used in document classification.

The Naive Bayesian classification model simply works on document classification as follows. Assume that a set of questioned documents needs to be classified into two classes, i.e., "spam" and "non-spam". The classification process starts by transforming a training dataset (i.e., "spam" and "non-spam" labeled documents) into feature vectors using a selected feature set. The resulted feature vectors are then used to train the Naive Bayesian classifier, i.e., the prior probabilities of two classes and the likelihood probabilities of documents given classes are estimated. Each questioned document is also represented as a vector using the same feature set used in representing the training dataset. Finally, the estimated probabilities (i.e., the prior and likelihood probabilities) are used to compute a posterior probability of each class given the feature vector of the questioned document. As the result, the question document is labeled with a class that maximizes the posterior probability.

Typically, in document analysis, a simple feature set that contains a group of words occurred in a document set is used to form feature vectors. The researchers normally call this type of features as *Bag of Words*. If we suppose that D words are selected to vectorise each document, then a vector of D elements is created for each document. Some approaches, such as Ye et al. (2009), Wajeed and Adilakshmi (2011) and Huang and Li (2011), have assigned the values of vector elements based on word frequency, where the value of each element in the vector represents the count of the corresponding word in the document. In other words, if it is assumed that x^d is a feature vector element referring to word d in the document and this word occurs exactly k times in the document, then the value of x^d is equal to k (i.e., $x^d = k$). k can be any value between 0 and K, where K is a maximum value of occurrence of all words of the feature set. Therfore, we can say that each word of the feature set can have a value of $K + 1$ categories. The naive Bayesian assumption is made that the probability of each word occurring in the document is totally independent of the occurrence of other words in the document. Since there are $K + 1$ possible categories for each element in the feature vector, the likelihood probability can be represented as a product of multinomials as shown in Eq 2.16.

$$p(x|y_j,\theta) = \prod_{d=1}^{D}\prod_{k=0}^{K}\theta_{djk}^{I(x^d=k)}, \tag{2.16}$$

where $\theta_{djk} = p(x^d = k|y_j)$ is the probability of class y_j to generate the feature x^d that occurs k times, and $I(x^d = k)$ is an indicator function equal to 1 if $x^d = k$ and 0 otherwise.

The Maximum Likelihood Estimation (MLE) of θ_{djk} is estimated as shown in Eq. 2.17

$$\theta_{djk}^{MLE} = \frac{N_{djk}}{\sum_{k'=0}^{K} N_{djk'}}, \tag{2.17}$$

where N_{djk} is the number of times that feature x^d (i.e., word d) occurs exactly k times in documents of class j.

In order to prevent zero counts in Eq. 2.17, the Laplace smoothing is used as shown in Eq. 2.18.

$$\theta_{djk}^{MLE} = \frac{N_{djk} + \alpha_k}{\sum_{k'=0}^{K} N_{djk'} + \alpha_{k'}}, \tag{2.18}$$

where α_k is a constant.

Some other approaches, such as Hussin and Kamel (2003), Koppel et al. (2011b) and Akiva and Koppel (2013), have suggested a simple way to assign the values vector elements based on a word occurrence, rather than word frequency. The value of each element in the vector (i.e., x^d, $d = 1, 2, \cdots, D$) is either 0 or 1, where 0 indicates that a corresponding word does not occur in the document, while 1 indicates that a corresponding word occurs in the document. Therefore, each feature vector forms a binary values vector and the likelihood probability can be represented as a product of Bernoulli distribution as shown in Eq. 2.19.

$$p(x|y_j,\theta) = \prod_{d=1}^{D}\theta_{dj}^{x^d}(1-\theta_{dj})^{1-x^d}, \tag{2.19}$$

where θ_{dj} is a probability of class y_j to generate the feature x^d.

The Maximum Likelihood Estimation (MLE) of θ_{dj} is estimated as follows.

$$\theta_{dj}^{MLE} = \frac{N_{dj}}{N_j}, \tag{2.20}$$

where N_{dj} is the number of times that feature x^d (i.e., word d) occurs in documents of class j and N_j is the number of documents of class j.

As well as, the Laplace smoothing technique is used in order to avoid zero counts as shown in Eq. 2.21.

$$\theta_{dj}^{MLE} = \frac{N_{dj} + \alpha_d}{N_j + \alpha}, \tag{2.21}$$

where α_d is a constant and $\alpha = \sum_d \alpha_d$.

In this Section, as shown above, it is assumed that the data are independently and identically distributed (iid) from an unknown probability distribution. However, in next Section, it is assumed that the data are sequential rather than iid.

2.3 Sequential Data: Hidden Markov Model

Different applications in our life have been modeled based on an assumption that data are independently and identically distributed (iid). For many applications, however, this assumption is deemed as a poor one (Rogovschi et al., 2010). Therefore, in this section, a different class of data sets, namely those that concern sequential data, will be addressed. The sequential data are often obtained through measurement of time series, or more generally, of sequence data, i.e., the order of the data is important. In fact, the sequential data contain a wealth of precious information because adjacent measurements and labels are expected to be related to each other in a form that can help to better grasp the underlying principles of many real-life problems.

The sequential data can be found in a variety of fields, such as pattern recognition, speech recognition and bio signal analysis. Typically, the Sequential data are modeled with Hidden Markov Model (HMM), which is defined as a dynamic classifier.

Before starting discussing the HMM and how it can be used for modeling the sequential data, it is important to briefly describe Markov models first.

2.3.1 Markov Models

The classical, easiest, way to represent N sequential data, $X = \{x_1, x_2, \cdots, x_N\}$, would be simply to disregard the sequential relations among data and assume that the data are independently and identically distributed (iid), as shown in Figure 2.4. However, this approach will fail to exploit the correlations among data.

FIGURE 2.4: A representation of N sequential data represented as independent, corresponding to a graph without links.

Therefore, it would be necessary to relax the iid assumption in order to utilize the correlations among data. The simplest way to do this is to consider a *Markov model* (also called *Markov chain*). The joint distribution of N data points (i.e., observations) using the Markov model can be defined using the product rule as follows.

$$p(x_1, x_2, \cdots, x_N) = \prod_{n=1}^{N} p(x_n | x_{n-1}, \cdots, x_1). \tag{2.22}$$

If it is assumed that each observation in the sequence is independent of all previous observations except for the most recent one, then the sequence is called the *first-order Markov chain*. Therefore, the joint distribution of N observations in first-order Markov chain is defined in Eq. 2.23.

$$p(x_1, x_2, \cdots, x_N) = p(x_1) \prod_{n=2}^{N} p(x_n|x_{n-1}). \tag{2.23}$$

That is, the conditional distribution of a present observation, x_n, given the sequence of all observations up to the present time, n, is given by

$$p(x_n|x_{n-1}, \cdots, x_1) = p(x_n|x_{n-1}), \qquad \textit{for all} \quad n \geq 2. \tag{2.24}$$

It is clear that only the value of the current observation will be used to predict the next observation in the sequence. The simplest first-order Markov chain is *stationary* (or *homogeneous*), where the conditional probabilities remain constant over time.

The graphical representation of the first-order Markov chain is represented in Figure 2.5.

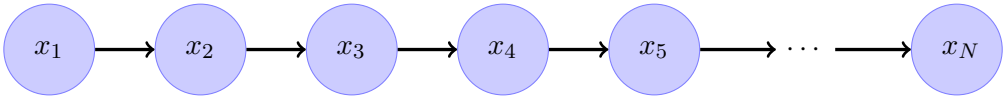

FIGURE 2.5: The first-order Markov chain.

For some applications, the values of the last two observations, rather than the last observation only, will provide useful information for predicting the value of the next observation, i.e., each observation is affected by two previous observations. Therefore, a *second-order Markov chain* is proposed. The joint distribution of N observations using the second-order Markov chain is defined in Eq. 2.25.

$$p(x_1, x_2, \cdots, x_N) = p(x_1)\ p(x_2|x_1) \prod_{n=3}^{N} p(x_n|x_{n-2}, x_{n-1}). \tag{2.25}$$

The graphical representation of the second-order Markov chain is represented in Figure 2.6.

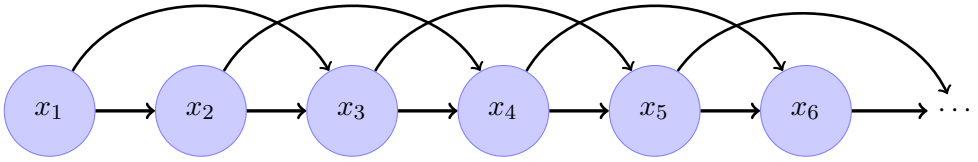

FIGURE 2.6: The second-order Markov chain.

In the same way, if each observation is affected by M previous observations, then an M^{th}-order Markov chain is presented.

2.3.2 Hidden Markov Model

The Hidden Markov Model (HMM) is a statistical probabilistic model used to form sequential data. The HMM is a popular technique used extensively in a variety of fields, such as speech recognition (Levinson et al., 1983; Juang and Rabiner, 1991), human actions (Yamato et al., 1992; Ahmad and Lee, 2006), handwriting recognition (Hu et al., 1996; Marti and Bunke, 2001), part-of-speech (POS) tagging of sentences (Brants, 2000; Collins, 2002), face recognition (Nefian and Hayes, 1998; Liu and Cheng, 2003) and bio signal analysis (Rabiner, 1989).

The HMM consists of a sequence of observable data and a hidden variable, which is not directly observable, for each observed data. The observable data are called "observations" and the hidden variables are called "hidden states". The hidden states in HMM form a Markov chain and the probability distribution of the observation depends on the underlying state.

Lets denote the N observations as $X = \{x_1, x_2, \ldots, x_N\}$ and the hidden states as $Q = \{q_1, q_2, \ldots, q_N\}$, where q_n is the hidden state of the n^{th} observation (i.e., x_n). Each observation, which is assumed to be a discrete symbol, has one value from the set of observations $W = \{w_1, w_2, \ldots, w_M\}$ and each hidden state has one value from the set of states $S = \{s_1, s_2, \ldots, s_T\}$. Here, M and T represent the number of distinct observations

and the number of distinct states in the model, respectively. Figure 2.7 illustrates the graphical structure of the HMM.

As shown in Figure 2.7, the hidden states, Q, form the first-order Markov chain where each state is independent of all previous states except for the most recent one, i.e.,

$$p(q_n|q_{n-1}, ..., q_1) = p(q_n|q_{n-1}), \qquad \textit{for all} \quad n \geq 2. \tag{2.26}$$

The conditional probability $p(q_n|q_{n-1})$, which shows how adjacent states are related, is called a "transition probability". The transition probabilities of all possible state values can be formed in an $T \times T$ transition matrix, denoted by $\boldsymbol{A}$. Each probability is given by $A_{ij} = p(q_n = s_j|q_{n-1} = s_i)$, where s_i, $s_j \in S$, $0 \leq A_{ij} \leq 1$ and $\sum_j A_{ij} = 1$.

The initial state q_1 , which is special in that it does not have a previous state, is defined as a marginal distribution $p(q_1)$. All initial states are represented by a $1 \times T$ vector, denoted by $\boldsymbol{\pi}$. Each probability is given by $\pi(i) = p(q_1 = s_i)$, where $s_i \in S$ and $\sum_i \pi(i) = 1$.

As seen in Figure 2.7, the probability of observation x_n depends only on the hidden state q_n. That means each observation is independent of all states and observations except for the hidden state that emits the considerable observation, i.e.,

$$p(o_n|q_N, x_N, \ldots, q_n, q_{n-1}, x_{n-1}, \ldots, , q_1, x_1) = p(x_n|q_n). \tag{2.27}$$

The conditional probability $p(x_n|q_n)$, which shows how observation x_n is related to hidden state q_n, is called the "emission probability". The conditional probabilities of all observations might, for example, be defined by Gaussians if the observations are continuous variables or by conditional probability matrices if the observations are discrete variables. In this thesis, the observations are assumed to be discrete symbols where each observation has one value of M possible values. Therefore, the emission probabilities of all observations given their states are formed in an $T \times M$ emission matrix, denoted by $\boldsymbol{B}$. Each conditional probability is given by $b_i(k) = p(x_n = w_k|q_n = s_i)$, where $w_k \in W$, $s_i \in S$.

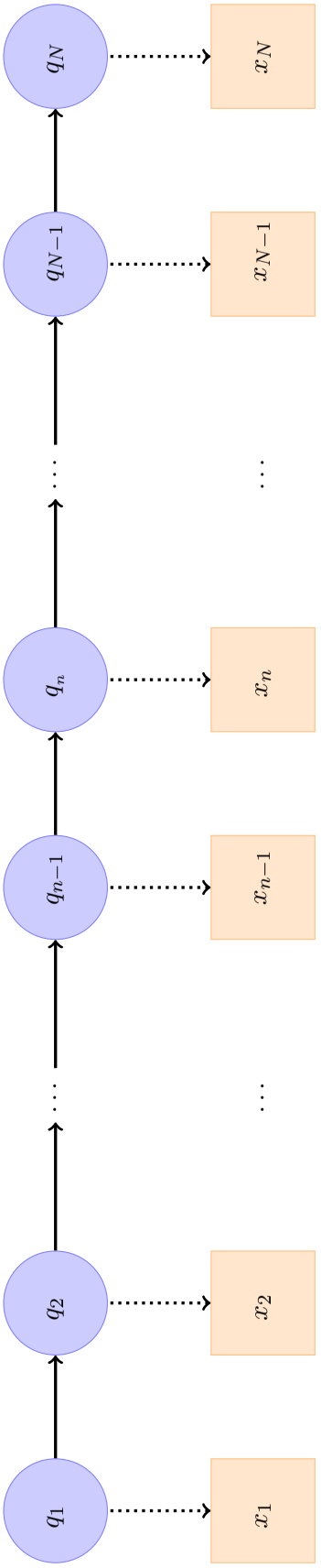

FIGURE 2.7: A graphical model of the HMM with N hidden states, $Q = \{q_1, q_2, \ldots, q_N\}$, and N observations, $X = \{x_1, x_2, \ldots, x_N\}$.

The transition and emission probabilities are assumed to be stationary conditional distribution, i.e., the distributions of $\boldsymbol{A}$ and $\boldsymbol{B}$ remain the same for all values of n.

Therefore, the HMM is defined by the above three probabilities, denoted as $\boldsymbol{\theta}$, with $\boldsymbol{\theta} = \{\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{\pi}\}$, for brevity.

The main objective of HMM is to find the best sequence of states (i.e., Q) that represents the sequence of observed data (i.e., X). In order to achieve that, the best estimate of HMM parameters, $\boldsymbol{\theta}$, is obtained first by using a forward-backward algorithm. Then, a decoding process, which aims to use the learned HMM parameters and find the best sequence of states for the corresponding observations, is performed using the Viterbi algorithm. The forward-backward algorithm and Viterbi algorithm are presented in the next subsections, respectively.

2.3.3 The Forward-Backward Algorithm

As illustrated in the previous subsection and seen in Figure 2.7, the HMM, which consists of sequence of hidden states (i.e., $Q = \{q_1, q_2, \ldots, q_N\}$) and independent observations (i.e., $X = \{x_1, x_2, \ldots, x_N\}$), can be specified by three parameters, $\boldsymbol{\theta} = \{\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{\pi}\}$. The model is learned by maximising the likelihood function of HMM in order to find a best estimation of $\boldsymbol{\theta}$ and so the probability of the observations becomes maximum, as in

$$\boldsymbol{\theta} = \arg\max_{\boldsymbol{\theta}} \left(p(X|\boldsymbol{\theta})\right). \tag{2.28}$$

The learning process of $\boldsymbol{\theta}$ is performed by using the *Baum-Welch algorithm* (Dempster et al., 1977), which is considered as a special case of the Expectation Maximisation (EM) algorithm. The process starts with using an initial value of $\boldsymbol{\theta}$ and computing the probabilities of being in each state at each time, which is done by using the *forward-backward algorithm* (Rabiner and Juang, 1986). After that, the estimated probabilities are used to obtain a better estimate of $\boldsymbol{\theta}$. Using the improved (hopefully) $\boldsymbol{\theta}$, the forward-backward algorithm is applied again and the cycle repeats until the convergence of either the $\boldsymbol{\theta}$ or the estimated probabilities is occurred.

The forward-backward algorithm is calculated using Eqs. 2.29 - 2.31.

$$p(q_i|X) = p(q_i|x_{1:i}, x_{i+1:N}). \tag{2.29}$$

$$p(q_i|x_{1:i}, x_{i+1:N}) \propto p(q_i, x_{1:i})p(x_{i+1:N}|q_i). \tag{2.30}$$

$$\gamma(q_i) \propto \quad \alpha(q_i) \qquad \beta(q_i). \tag{2.31}$$

One can see that the forward-backward algorithm $\gamma(q_i)$ is a combination of two probability algorithms. The first probability algorithm $\alpha(q_i)$, which passes forward in order, is called forward algorithm, while the second probability algorithm $\beta(q_i)$, which passes backward in order, is called backward algorithm.

The forward algorithm represents the joint probability of observing all of the previous observations up to the observation i and the value of state q_i. The forward algorithm is represented in Eq. 2.32.

$$\alpha(q_i) = p(x_i|q_i) \sum_{q_{i-1}} p(q_i|q_{i-1})\alpha(q_{i-1}). \tag{2.32}$$

According to Eq. 2.32, the iteration starts at finding the join probability of the first observation and the first state, which is:

$$\alpha(q_1) = p(x_1, q_1) = p(q_1)p(x_1|q_1). \tag{2.33}$$

On the other hand, the backward algorithm represents the conditional probability of all future observations starting from observation $i + 1$ up to T given the state q_i. The backward algorithm is represented in Eq. 2.34.

$$\beta(q_i) = \sum_{q_{i+1}} p(x_{i+1}|q_{i+1})p(q_{i+1}|q_i)\beta(q_{i+1}). \tag{2.34}$$

One can see that the recursion starts by finding the conditional probability of the last observation given the last state, $\beta(q_N)$. The value of this conditional probability is equal to 1 for all settings of q_N.

The learned $\boldsymbol{\theta}$ will be used in the next subsection in order to find the best state sequence that represents the observation sequence.

2.3.4 The Viterbi Algorithm

The main goal of HMM is to find the most likely sequence of states (i.e., Q) that generate the corresponding sequence of observations (i.e., X), as shown in Eq. 2.35. Actually, the issue of finding the optimal sequence of states is not the same as that of finding the individual optimal states. The second issue can be solved using the forward-backward algorithm by finding the state variable marginal $\gamma(q_i)$ and then maximising each of these separately (Duda et al., 2001).

$$Q^* = \arg\max_{Q} p(Q|X). \tag{2.35}$$

According to the Naive-Bayesian model, Eq. 2.35 can be re-expressed as:

$$Q^* = \arg\max_{Q} \frac{p(Q,X)}{p(X)}. \tag{2.36}$$

The value of $p(X)$ does not have any effect on the maximisation process and Eq. 2.36 can be then re-written as:

$$Q^* = \arg\max_{Q} p(Q,X) = \arg\max_{q_1,\ldots,q_N} p(q_1,\ldots,q_N,x_1,\ldots,x_N). \tag{2.37}$$

In fact, the number of potential routes through a sequence of states in HMM increases exponentially with the length of the sequence. Therefore, the *Viterbi algorithm* (Viterbi, 1967), also known as *max sum algorithm*, is used in order to find efficiently the most likely sequence of states for the given observations in HMM, where the number of potential routes increases only linearly, rather than exponentially, with the length of the sequence.

In the Viterbi algorithm, the probability of the most likely sequence ending in state $q_n = s_j$, where $s_j \in S$, given the first n observations is defined and denoted by $\delta_n(j)$. This probability can be computed recursively using the Eqs. 2.38-2.42.

$$\delta_n(j) = \max_{q_1,\ldots,q_{n-1}} p(q_1, \ldots, q_{n-1}, q_n = s_j, x_1, \ldots, x_n). \tag{2.38}$$

$$\delta_n(j) = \max_{q_1,\ldots,q_{n-1}} p(x_n|q_n = s_j)\ p(q_n = s_j|q_{n-1})\ p(q_1, \ldots, q_{n-1}, x_1, \ldots, x_{n-1}). \tag{2.39}$$

$$\delta_n(j) = \max_{q_{n-1}} p(x_n|q_n = s_j)\ p(q_n = s_j|q_{n-1}) \max_{q_1,\ldots,q_{n-2}} p(q_1, \ldots, q_{n-1}, x_1, \ldots, x_{n-1}). \tag{2.40}$$

$$\delta_n(j) = \max_{i=1,\ldots,T} p(x_n|q_n = s_j)\ p(q_n = s_j|q_{n-1} = s_i) \max_{q_1,\ldots,q_{n-2}} p(q_1, \ldots, q_{n-1} = s_i, x_1, \ldots, x_{n-1}). \tag{2.41}$$

$$\delta_n(j) = \max_{i=1,\ldots,T} p(x_n|q_n = s_j)\ p(q_n = s_j|q_{n-1} = s_i)\ \delta_{n-1}(i). \tag{2.42}$$

Furthermore, the most likely value of the state q_{n-1} which leads to $q_n = s_j$ is defined as:

$$\Psi_n(j) = \arg\max_{i=1,\ldots,T} p(q_n = s_j|q_{n-1} = s_i)\ \delta_{n-1}(i). \tag{2.43}$$

The iterative process shown in Eq. 2.42 is terminated by finding the most probable last state (i.e., $n = N$) of the most likely sequence of states, as shown in Eq. 2.44.

$$q_N = \underset{i=1,\ldots,T}{\arg\max}\ \delta_N(i). \tag{2.44}$$

For the other states (i.e., $n \neq N$), a backtracking is performed in order to find the most probable state of the most likely sequence of states. This is done using Eq. 2.45.

$$q_n = \Psi_{n+1}(q_{n+1}) \qquad n = N-1, \ldots, 1. \tag{2.45}$$

After all, the best sequence of states, $Q^* = \{q_1, q_2, \ldots, q_N\}$, that represents the corresponding observations is determined.

2.3.5 Hidden Markov Model in Document Analysis

Due to the unique features and properties of Hidden Markov Model (HMM) in handling sequential patterns, the HMM is of great practical importance for many linguistic applications and problems. One of these problems that has been intensely investigated during the past several years is document analysis. Document analysis is primarily concerned with reviewing and evaluating documents in order to find, select, appraise (i.e., making sense of) and synthesize data contained within the documents (Bowen, 2009). Many researchers have exploited the contextual information hidden between characters, words, sentences or passages through using of HMM in order to perform different tasks of document analysis. For example, the works of Kupiec (1992) and Stratos et al. (2016) have used a HMM for the task of part-of-speech tagging, i.e., the task of assigning each word in a sentence a tag that describes how that word is used in the sentence. Furthermore, in Thede and Harper (1999), the authors have employed a second-order HMM for the same task.

Some researchers, such as Yamron et al. (1998) and Blei and Moreno (2001), have modeled a HMM for the task of document topic segmentation, where a document is considered as mutually independent sets of words generated by a latent topic variable

in a time series. Some other researchers, such as Denoyer et al. (2001), have developed HMM-besed approaches for supervised document classification and ranking with respect to category. Furthermore, in Nikolaos and George (2008) and Yi and Beheshti (2013), the authors have used HMMs to automatically categorize digital documents into a standard library classification schema. The work of Vieira et al. (2014) has also used a HMM for classifying biomedical scientific documents according to their content.

Many works of document analysis have been done on analysing spoken documents using the HMM. For example, the works of Chen et al. (2006) and Maskey and Hirschberg (2006) are for building a HMM for the task of spoken document summarization (i.e., identifying information from a spoken document that summarizes, or captures the essence of the document). In these works, the HMM is applied in order to predict the optimal sequence of sentences that best summarize the spoken document.

In Pinto et al. (2003) and e Silva (2009), the authors have employed an HMM in order to find table regions in a text. In addition, the work of Southavilay et al. (2010) has implemented a HMM-based technique for extracting a semantic meaning of writing activities and analysing the writing processing of collaborative documents written by groups of students. Another task in document analysis that can avail the features of HMM is multi-author document decomposition according to authorship, which is the task addressed in this thesis. The main idea is to make use of the contextual relationship between sentences in order to capture the authors of the sentences.

The Naive Bayesian, which is presented in Section 2.2, and Hidden Markov Model (HMM), which is presented in Section 2.3, are employed mainly in a supervised learning to classify a set of data, i.e., the classes are predefined. In next section (i.e., Section 2.4), a clustering method for unsupervised learning, where the classes are not predefined, will be presented.

2.4 Clustering Methods: Gaussian Mixture Models

In statistics and machine learning, the problem in which there are no labeled data used to train a model, like the problem we are dealing with in this thesis, is called

unsupervised learning. One of the most commonly adopted strategies in unsupervised learning is data clustering. The data clustering is the process of segmenting unlabeled data into groups, or clusters, that are not previously defined. Several techniques have been developed for data clustering, such as k-means clustering (MacQueen et al., 1967), Spectral clustering (Von Luxburg, 2007) and Fuzzy c-means clustering (Bezdek et al., 1984). In this section, another technique for data clustering is presented and discussed. The technique is Gaussian Mixture Models (GMMs). Before starting talking about the GMMs, a brief discussion of a Gaussian distribution is given.

2.4.1 The Gaussian Distribution

The Gaussian distribution, also named as the normal distribution, is a commonly used model for the distribution of continuous variables. The Gaussian distribution is undoubtedly one of the most famous and useful distribution in statistics and plays a significant role in numerous applications in engineering, physics and many other fields (Nandi and Mämpel, 1995; Kumar et al., 2010).

The univeriate Gaussian distribution of a one-dimensional feature vector x is formed as shown in Eq. 2.46.

$$\mathcal{N}(x|\mu,\sigma^2) = \frac{1}{(2\pi\sigma^2)^{1/2}}\ exp\ \left\{-\frac{1}{2\sigma^2}(x-\mu)^2\right\}, \tag{2.46}$$

where μ and σ^2 are the mean and the variance of the univeriate Gaussian distribution.

The multivariate Gaussian distribution of a D-dimensional feature vector x is formed as shown in Eq. 2.47.

$$\mathcal{N}(x|\mu,\Sigma) = \frac{1}{(2\pi)^{D/2}}\frac{1}{|\Sigma|^{1/2}}\ exp\ \left\{-\frac{1}{2}(x-\mu)^T\ \Sigma^{-1}\ (x-\mu)\right\}, \tag{2.47}$$

where μ is a D-dimensional mean vector, Σ is a $D \times D$ covariance matrix and $|\Sigma|$ is the determinant of Σ.

2.4.2 Mixtures of Gaussians

In statistics, a case where a mixture of multiple component distributions that collectively make a mixture distribution is called mixture model. The mixture model of K component distributions of a vector x is formed as follows.

$$p(x) = \sum_{k=1}^{K} \alpha_k \; p_k(x). \tag{2.48}$$

The α_k is a mixing coefficient for the k^{th} component. The α_k must satisfy

$$0 \leqslant \alpha_k \leqslant 1. \tag{2.49}$$

together with

$$\sum_{k=1}^{K} \alpha_k = 1. \tag{2.50}$$

The $p_k(x)$, which is shown in Eq. 2.48, can be one of statistical distributions employed to model the data. If the $p_k(x)$ is substituted with the multivariate Gaussian distribution, then the mixture model, $p(x)$, will represent the Gaussian mixture models (GMMs), which is presented in Eq. 2.51.

$$p(x) = \sum_{k=1}^{K} \alpha_k \; \mathcal{N}(x|\mu_k, \Sigma_k). \tag{2.51}$$

The model is a weighted linear superposition of K multivariate Gaussian distributions. Each one is characterized by a multivariate normal distribution with weight α_k, mean μ_k and a covariance matrix Σ_k for $k = 1, 2, \cdots, K$.

The clustering process using GMMs is performed by assigning the data to the multivariate normal components that maximise the component posterior probability given the data. The GMMs are trained using the iterative Expectation-Maximization (EM) algorithm. In particular, the EM algorithm, which is discussed in the next subsection,

aims to find maximum likelihood function of data with respect to the model parameters in an efficient way.

2.4.3 Expectation-Maximisation for GMMs

Suppose that there is a set of N data observations, $\mathbf{x} = \{x_1, x_2, \cdots, x_N\}$ that need to be clustered using GMMs of K components. The Expectation-Maximization (EM) algorithm (Dempster et al., 1977; Bilmes et al., 1998) is used to train the model parameters (i.e., μ_k, Σ_k and α_k for $k = 1, 2, \cdots, K$) by maximising the likelihood function of data.

In general, the EM algorithm is an iterative process that consists of an *E-Step* and an *M-Step*. The algorithm starts from the initial estimate of the model parameters, which is often done randomly, and evaluates the initial value of the log likelihood of data. The *E-Step* of the EM algorithm evaluates the following probability, which is sometimes named as responsibility (Bishop, 2006), for all data points, $\mathbf{x}$, and all K components using the current values of the model parameters.

$$\gamma_{nk} = \frac{\alpha_k \, \mathcal{N}(x_n|\mu_k, \Sigma_k)}{\sum_{j=1}^{K} \alpha_j \, \mathcal{N}(x_n|\mu_j, \Sigma_j)}, \tag{2.52}$$

where x_n is a D-dimensional feature vector, α_k is a mixing coefficient for the k^{th} component and $\mathcal{N}(x_n|\mu_k, \Sigma_k)$ is a multivariate Gaussian distribution with mean μ_k and covariance matrix Σ_k.

The probability γ_{nk} is computed for all data points (i.e., $n = 1, 2, \cdots, N$) and all mixture components (i.e., $k = 1, 2, \cdots, K$), where $\sum_{k=1}^{K} \gamma_{nk} = 1$.

The *M-Step* of the EM algorithm re-estimates the model parameters using the current responsibilities as follows.

$$\alpha_k^{new} = \frac{N_k}{N}, \qquad \text{For } k = 1, 2, \cdots, K. \tag{2.53}$$

$$\mu_k^{new} = \frac{1}{N_k} \sum_{n=1}^{N} \gamma_{nk}\ x_n, \qquad \text{For } k = 1, 2, \cdots, K. \tag{2.54}$$

$$\Sigma_k^{new} = \frac{1}{N_k} \sum_{n=1}^{N} \gamma_{nk}(x_n - \mu_k^{new})(x_n - \mu_k^{new})^T, \qquad \text{For } k = 1, 2, \cdots, K. \tag{2.55}$$

The value of N_k is defined as the effective number of data points assigned to component k. That is,

$$N_k = \sum_{n=1}^{N} \gamma_{nk}. \tag{2.56}$$

The new model parameters are now ready to compute the log likelihood of the data using Eq. 2.57.

$$\text{In } p(\mathbf{x}|\mu, \Sigma, \alpha) = \sum_{n=1}^{N} \text{In} \left\{ \sum_{k=1}^{K} \alpha_k \mathcal{N}(x_n|\mu_k, \Sigma_k) \right\}. \tag{2.57}$$

The *E-Step* and *M-Step* of the algorithm are then repeated until the convergence of either the log likelihood of the data or the values of the model parameters is achieved.

2.4.4 Gaussian Mixture Models in Document Analysis

Gaussian Mixture Models (GMMs) are a probabilistic model which has been successfully applied in numerous studies of document analysis. In fact, the GMMs technique plays an important role in data clustering with a focus on density estimation and pattern recognition because of its mathematical tractability, asymptotic properties and computational convenience. The GMMs technique has also the ability to model any given probability distribution function when the number of Gaussian components is large enough (Xian et al., 2015).

Traditionally, the main task of GMMs in document analysis is to cluster a set of documents (or text) based on specific criteria. For example, the authors of Liu et al. (2002) have used the GMMs for clustering a group of documents according to topic and main

content. Furthermore, the authors of Xing et al. (2014) have employed the GMMs on document classification task.

The works reported in Schlapbach and Bunke (2006), Schlapbach and Bunke (2008) and Christlein et al. (2014) have applied the GMMs in writer identification task, i.e., determining the author of a handwriting document from a set of writers. In these works, the distribution of feature vectors extracted from a person's handwriting is modeled by a Gaussian mixture density.

2.5 Summary

This chapter has presented a background information of authorship analysis and briefly discussed some of existing techniques that have been used in authorship analysis. The stylometric feature sets and feature representations that are engaged in representing text and capturing writing styles are varied. The most three common stylometric feature sets and feature representations used in authorship analysis have been listed in this chapter.

Several tasks are included in authorship analysis. One of these tasks is an authorship-based multi-author document decomposition, which is the topic addressed in this thesis. The authorship-based multi-author document decomposition is the task for segmenting sentences of a multi-author document into authorial components. Each component contains the sentences that are written by the same author. In this chapter, some of the difficulties associated with the task have been explored.

Two different scenarios will be considered in this thesis to handle the authorship-based multi-author document decomposition task. The first scenario is when an independency between the authority of sentences is assumed, i.e., it is assumed that knowing the author of a sentence in a document does not have any relation with the authors of other sentences in the document. Therefore, in this thesis, a probabilistic-based model for independent data (i.e., Naive Bayesian model) is employed to address this scenario. However, the second scenario is when a sequential data between the authority of sentences is assumed, i.e., it is assumed that knowing the author of an sentence in a document does have a relation with the authors of other sentences in the document. Therefore, in this thesis,

a probabilistic-based model for sequential data (i.e., Hidden Markov Model (HMM)) is developed to address this scenario. In this chapter, the Naive Bayesian classifier and the HMM both have been described.

This thesis, as mentioned above, addresses the task of authorship-based multi-author document decomposition. This task is considered as an example of unsupervised learning algorithm where no desired outputs are available. Therefore, in this thesis, a clustering technique (i.e., Gaussian Mixture Models (GMMs)) is applied for the unsupervised learning algorithm. This chapter has presented and reviewed the GMMs clustering technique and its applications in document analysis.

Chapter 3

Unsupervised Decomposition of a Multi-Author Document Based on Naive-Bayesian Model

This chapter proposes a new unsupervised approach for decomposing a multi-author document into authorial components. We assume that we do not know anything about the document and the authors, except for the number of the authors of that document. The key idea is to exploit the difference in the posterior probability of the Naive-Bayesian model to increase the precision of the clustering assignment and the purity result of our approach. Even if this type of problem has a great attention on security and forensic investigation, there are very few researches trying to think through it. We systematically evaluate the efficiency of our proposed approach by using three benchmark datasets. An authentic document is also used to examine the performance of the proposed approach on authentic documents. Experimental results show that the proposed approach outperforms three state-of-the-art approaches.

3.1 Introduction

The traditional studies on text segmentation, as shown in Choi (2000), Brants et al. (2002), Misra et al. (2009) and Hennig and Labor (2009), focus on dividing the text into significant components such as words, sentences and topics rather than authors. Those works were done based on Natural Language Processing (NLP) techniques and various machine learning schemas. Nowadays, due to the availability of online communication facilities, the cooperation between authors to produce a document becomes much easier. The co-authored documents include Web pages, books, academic papers and blog posts. There are a very few approaches that have concentrated on developing techniques for segmentation of a multi-author document according to the authorship. As a matter of fact, many applications can take advantage of these techniques. For example, the techniques can be used to clarify the contributions of the individual authors of a document. The techniques can also be applied in security reasons such as forensic investigation (Iqbal et al., 2008) and plagiarism detection (Zu Eissen et al., 2007; Stamatatos, 2009b). Most of existing approaches that are closely related to the research of this thesis, such as those in Schaalje et al. (2013), Layton et al. (2013) and Segarra et al. (2014), have dealt with determining authors of single-author documents. The works of Rosen-Zvi et al. (2004), Rosen-Zvi et al. (2010) and Savoy (2013a), however, have worked on segmenting a group of single-author documents according to topics in order to extract information about their authors.

In Koppel et al. (2011a), the authors have developed an approach for segmenting a multi-author document according to authorship. The approach requires manual translations and concordance to be available beforehand. Hence, the approach can only be applied on particular types of documents such as Bible books. Akiva and Koppel (2012) have investigated this limitation and presented a generic unsupervised approach for multi-author document segmentation according to authorship. However, the performance of segmentation is not good enough. The authors of Akiva and Koppel (2013) have developed an approach which relies on the distance measurement to increase the precision and accuracy of the clustering and classification process. Two different sets of features have been used in the approach in order to capture authors' writing styles. The first

feature set contains 500 most common words in the document. The second feature set, which is only valid on special types of documents like Bible books, contains 1595 synonyms written in Hebrew language. The performance of the approach is degraded when the number of authors increases to more than two.

The contributions of this chapter are as follows.

- A procedure for segment elicitation is developed and it is applied in the clustering assignment process. It is for the first time to develop such a procedure relying upon the differences in the posterior probabilities.
- A probability indication procedure is developed to improve the purity of sentence segmentation. It selects the significant and trusted sentences from a document and involves them to relabel all sentences in the document. Our approach does not require any information about the document and the authors other than the number of authors of the document. Therefore, it is completely unsupervised learning.
- Our proposed approach is not restricted to any type of documents. It is still workable even when the topics in a document are not detectable.

The organisation of this chapter is as follows. Section 3.2 demonstrates the framework of the proposed approach. Section 3.3 discusses the segmentation process, feature representation and clustering techniques which are used in our approach. Section 3.4 demonstrates the segment elicitation procedure and feature re-vectorization. Section 3.5 illustrates a supervised learning process. Section 3.6 describes a probability indication procedure. Experimental results are conducted in Section 3.7. Finally, Section 3.8 presents the summary of the chapter.

3.2 Framework of the Proposed Approach

Precisely, the problem that we are interested in can be formulated as follows. Given a multi-author document written by N authors, it is assumed that every author has

written consecutive sequences of sentences, and every sentence is completely written by only one of the N authors. The value of N is pre-defined. The objective is to segment the sentences in the document into authorial components.

Our approach goes through the following steps:

- *Step 1* Divide the document into segments of fixed length.
- *Step 2* Represent the resulted segments as vectors using an appropriate feature set which can differentiate the writing styles among authors and make a distinction between them.
- *Step 3* Cluster the resulted vectors into N clusters using an appropriate clustering algorithm targeting on achieving high *recall* rates.
- *Step 4* Re-vectorize the segments using a different feature set to more accurately discriminate the segments in each cluster.
- *Step 5* Apply the "*Segment Elicitation Procedure*" to select the best segments from each cluster to increase the *precision* rates.
- *Step 6* Re-vectorize all selected segments using another feature set that can capture the differences among the writing styles of all sentences in a document.
- *Step 7* Train the classifier using the Naive-Bayesian model.
- *Step 8* Label each sentence in the document using the learned classifier.
- *Step 9* Apply the "*Probability Indication Procedure*" to increase the *purity* of the sentence classification process using five criteria.

The framework of the proposed approach is shown in Figure 3.1.

To assess the performance of the proposed scheme, we perform our experiments on an artificially merged document. The artificially merged document are generated by employing the same procedure used in Koppel et al. (2011a), Akiva and Koppel (2012) and Akiva and Koppel (2013) for fair comparison. This procedure aims to combine a group of documents of N authors into a single merged document. Each of these

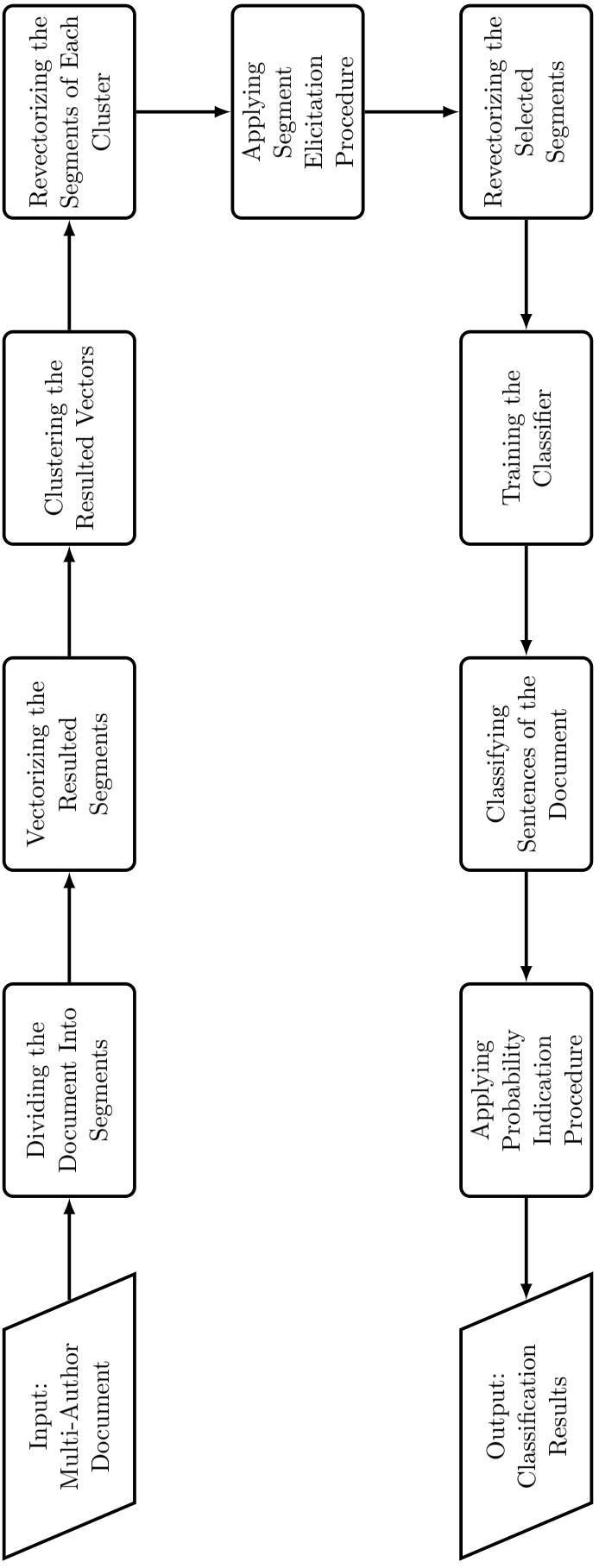

FIGURE 3.1: The framework of the proposed approach

documents is written by only one author. The generation of the merged document begins with randomly choosing an author from an author list. Then, we pick up the first m previously-unselected sentences from a document of that author, and merge them with the first m previously-unselected sentences from the documents of other randomly selected authors. This continues until all sentences from all authors' documents are selected. The value of m on each switch is an integer value chosen randomly from a uniform distribution varying from 1 to V. That means that the sentences of each author are distributed randomly over the document, and this makes our problem more factual and more difficult. We also follow the approaches of Koppel et al. (2011a), Akiva and Koppel (2012) and Akiva and Koppel (2013) and assign 200 to V.

For interpretative purpose, we will exploit the bible books of Ezekiel and Job to create a merged document. The bible book of Ezekiel contains 1,273 sentences and the bible book of Job contains 1,018 sentences. We use this example of a merged document to clarify each step of our proposed framework. We also use this merged document to work out the values of parameters used in our approach. We create the merged document using the procedure described above. In the merged document, there are 2,291 sentences in total and there are hence 20 transitions from Ezekiel sentences to Job sentences and from Job's to Ezekiel's.

3.3 Segmentation, Feature Representations and Clustering

For *Step 1* of our approach shown in Section 3.2, we divide the merged document into segments. Each segment has v sentences. The value of parameter v is chosen in a way that the division of the document can generate segments with a sufficient length to reflect the authors' writing styles, and also, it should result in an adequate number of segments (so as to be employed later to form a training dataset for the Naive Bayesian classifier). In the Ezekiel-Job merged document, we set the value of v to be 30 because we get the highest purity result with this value as shown in Table 3.2. Table 3.2 shows the purity result obtained by applying our approach using different values of v on the

TABLE 3.1: The clustering results of segments in the Ezekiel-Job document

Segments	**Cluster 1**	**Cluster 2**
Ezekiel	34	0
Job	0	27
Mixed	8	8
Total	**42**	**35**

Ezekiel-Job merged document. Table 3.2 illustrates also the influence of changing the value of v on the purity results. A large value of v (indicating long segments) does not give a preferable result since the number of segments is not enough for training a supervised learning in this case. On the other hand, a small value of v (corresponding to short segments) does not produce superior result since the length of segments is not adequate for representing the authors' writing styles in this case. As a result, we get 77 segments of 30 sentences each (except for the last segment which has only 11 sentences). In these 77 segments, 34 segments are written by Ezekiel, 27 segments are written by Job and 16 are mixed segments written by Ezekiel and Job.

For *Step 2* of our approach, we represent each segment using a binary vector that reflects all words that appear at least three times in the document. In the Ezekiel-Job merged document, we represent all 77 segments using a binary vector that reflects all words that appear at least three times in the document.

In *Step 3*, the Gaussian Mixture Models (GMMs) are applied in order to cluster the segments (i.e., feature vectors) to N multivariate Gaussian densities. The GMMs are trained using the iterative Expectation-Maximization (EM) algorithm. More details about the GMMs can be found in Section 2.4. In the Ezekiel-Job document, the 77 segments (34 Ezekiel's segments, 27 Job's segments and 16 mixed segments) are clustered by using the GMMs into two multivariate Gaussian densities. We set the value of N to be two (Ezekiel and Job). The results of this clustering are shown in Table 3.1.

We find that all 34 Ezekiel segments are clustered in Cluster 1 (Ezekiel cluster), and all 27 Job segments are clustered in Cluster 2 (Job cluster). Mixed segments are divided equally between the two clusters. Note that the *recalls* of both clusters are 100%, and the precisions are 81% and 77% in Cluster 1 and Cluster 2, respectively.

TABLE 3.2: Purity results of applying our approach in the Ezekiel-Job document using different values of segment length (v) and different values of vital segments percentage (s)

		Vital Segments percentage (s)									
		10%	**20%**	**30%**	**40%**	**50%**	**60%**	**70%**	**80%**	**90%**	**100%**
Segment length (v)	**5**	84.2%	88.7%	88.9%	87.8%	86.3%	86.4%	86.9%	87.1%	86.3%	85.0%
	10	85.6%	90.7%	91.2%	90.1%	88.5%	88.7%	89.6%	90.6%	90.8%	90.0%
	15	88.9%	92.1%	93.4%	90.5%	90.0%	91.2%	92.8%	93.2%	93.5%	94.3%
	20	81.2%	96.2%	97.9%	98.1%	96.6%	97.5%	98.2%	98.5%	98.7%	98.7%
	25	82.1%	97.2%	97.7%	94.1%	91.4%	93.0%	92.2%	92.1%	94.1%	95.2%
	30	87.4%	95.2%	95.4%	96.0%	96.9%	96.8%	97.0%	**99.0**%	97.4%	97.7%
	35	83.0%	91.3%	96.1%	96.5%	96.6%	96.5%	96.6%	97.7%	98.0%	97.6%
	40	63.7%	72.9%	80.3%	85.2%	84.9%	85.0%	86.9%	86.3%	87.3%	87.2%
	45	55.4%	73.1%	97.2%	98.5%	98.8%	98.5%	98.5%	98.7%	98.3%	97.6%
	50	57.8%	55.7%	60.1%	82.0%	82.8%	86.0%	88.3%	87.5%	88.3%	87.7%
	55	53.6%	53.8%	67.9%	81.1%	85.4%	85.1%	86.0%	86.9%	86.8%	82.3%

Based on the Gaussian component that a segment is clustered to during the above clustering process, each segment is given a label. The segments of each class (noting that, the term 'cluster' is now substituted with 'class') will be filtered in the next section by applying the proposed Segment Elicitation procedure. The filtered segments will be used later to form a training dataset for the Naive Bayesian classifier.

3.4 Segment Elicitation Procedure and Feature Re-vectorization

For *Step 4* of our approach shown in Section 3.2, all of the segments in all clusters are re-vectorized using the binary representation of the 1500 most frequently-appeared words in the document.

For *Step 5* of our approach, a Segment Elicitation procedure is proposed. The key idea is to choose only the segments from a cluster that can best represent the writing style of the cluster. We call these selected segments *vital segments*.

The vital segments have the following two features. First, they can represent the expressive style of a specific cluster. Second, they can distinguish the writing style of that cluster from other clusters. After all, the purpose of this procedure is to find the vital segments of each cluster to form a training dataset for the supervised learning in Step 7 of the approach.

To find the vital segments of each class, we consider the differences in the posterior probabilities of each segment according to the other classes. The posterior probability of segment x in class c_i, where $i \in \{1, 2, \cdots, N\}$, is computed using the Naive-Bayesian model as shown in Eq. 3.1. We assume that the features are mutually independent. More details about the Naive-Bayesian model can be found in Section 2.2.

$$p(c_i|x) = \frac{p(c_i)p(x|c_i)}{p(x)}, \qquad i \in 1, 2, \cdots, N, \tag{3.1}$$

where the likelihood probability of the segment, $p(x|c_i)$, is estimated as follows.

$$p(x|c_i) = \prod_{d=1}^{D} p(x^j|c_i). \tag{3.2}$$

For a given class c_i, $i \in \{1, 2, \cdots, N\}$, if any one of the D features of segment x does not exist, then the probability value in Eq. 3.2 becomes zero. In order to avoid this probability value from becoming zero, the Laplace Smoothing as shown in Eq. 3.3 is used to regularize this value:

$$p(x^j|c_i) = \frac{n_{di} + 1}{n_i + D}\ , \tag{3.3}$$

where n_{di} is the number of times that feature x^d (i.e., word d) occurs in a segments of class i, n_i is the number of segments of class i and D is the dimension of the segment's features.

The computation of the likelihood probability of segment x in Eq. 3.2 needs D multiplication operations. Note that $p(x|c_i)$ has a value between 0 and 1. Floating point underflow may occur when the multiplication operations are performed. Therefore, we change the multiplication operations to addition operations by taking logarithms on both sides of Eq. 3.2. We do the same to Eq. 3.1.

For a given segment, its posterior probability $p(c_i|x)$ of a class provides a good indication in determining the importance of the segment in the class. However, there are some segments which posterior probabilities are high in more than one class, so these segments cannot be used to infer which class the segments should belong to and hence are not good features to distinguish between the classes. Therefore, we proceed to remove this type of segments to obtain the vital segments based on their posterior probabilities.

For each segment x in a class, the difference between its posterior probability of the class and its maximum posterior probability of all other classes is computed. We are looking for segments that have high posterior probabilities of their classes and have low posterior probabilities of other classes. We select $s\%$ of the segments from each class as vital segments that have the biggest differences mentioned above. Here, the percentage of the segments to be selected in each class, s, should be chosen carefully for the optimal

extraction of vital segments and feature representation of a class. Furthermore, the number of vital segments should be large enough to form a training dataset for the supervised learning in Step 7.

In the Ezekiel-Job document, Cluster 1 is the Ezekiel class and Cluster 2 is the Job class. There are 42 segments assigned into the Ezekiel class, of which only 34 of them (i.e. 81%) are correctly assigned and 8 are the segments mixed from the Ezekiel and Job documents. There are 35 segments assigned to the Job class, of which only 27 of them (i.e. 77%) are correctly assigned and 8 are the segments mixed from the Ezekiel and Job documents. From each class, 80% of the segments that have the biggest differences are selected and used as vital segments. Table 3.2 lists the final purity results using different percentage values for testing on the Ezekiel-Job document. Table 3.2 shows that 80 percent produces the highest purity. As a result, we get 34 vital segments for the Ezekiel class and 28 vital segments for the Job class. Of the 34 vital segments in Ezekiel class, 30 are truly written by Ezekiel, and of the 28 vital segments in Job class, 25 are truly written by Job. As a result, the precisions of Ezekiel class and Job class are increased to 88.2% and 89.3%, respectively.

The vital segments for two classes are used in the next section to train the supervised classifier which can best classify each sentence to the correct author's class.

3.5 Supervised Learning

For *Step 6* of our approach shown in Section 3.2, the vital segments are represented in terms of the frequencies of all words that have appeared at least three times in the whole document.

In *Step 7*, the Naive-Bayesian model is applied to learn a classifier. The goal of the classifier is to classify each sentence in the document into one of the N classes. The classifier is trained using the vital segments for the N classes.

In *Step 8*, the classifier is used to predict the class label c_i of each sentence in the document. In the Ezekiel-Job document, the Naive-Bayesian model is used to learn a

classifier. We use the 62 vital segments for the two classes (i.e., the Ezekiel and Job classes) to train the supervised classifier.

The learned classifier is used to classify all sentences in the merged document into the Ezekiel class or the Job class. We find that 93.2% of all sentences of Ezekiel and Job classes are correctly classified. The results of the classification are shown in Table 3.3.

TABLE 3.3: The purity results of sentences in the Ezekiel-Job document

	Ezekiel Class	**Job Class**
Ezekiel Sentences	1,208	65
Job Sentences	92	926

As shown in Table 3.3, the purity results are not satisfactory enough because there are still 157 misclassified sentences in total. We have observed that the misclassification is mainly due to the following two reasons.

1. Some sentences may not have sufficiently discriminative features to be correctly classified into a class.

2. There may be a case that one sentence shares some features used to classify two or more classes so that the posterior probabilities of these classes for the sentence are close. These sentences may not have been classified with a high confidence and have hence led to a misclassification.

In the next section, a probability indication procedure is presented and applied in order to enhance the purity result of the Naive-Bayesian classifier.

3.6 Probability Indication Procedure

For *Step 9* of our approach shown in Section 3.2, a probability indication procedure is proposed based on the following five criteria.

1. Any sentence in the document is considered as *trusted sentence* if its posterior probability in its class is greater than its posterior probabilities in all other classes by more than a threshold q. Thereupon, every trusted sentence holds its class.

2. If the first sentences in the document are not deemed to be trusted sentences, then they are assigned to the same class of the first trusted sentence that follows them. Figure 3.2 illustrates this criterion.

3. If the last sentences in the document are not deemed to be trusted sentences, then they are assigned to the same class of the last trusted sentence that precedes them. Figure 3.2 illustrates this criterion.

4. If a group of unassigned sentences is located between two trusted sentences which have the same class, then all of the sentences in that group are assigned to the same class of these trusted sentences. Figure 3.3 illustrates this criterion.

5. If a group of unassigned sentences is located between two trusted sentences which have different labels, then the best separating point in that group is detected to separate it into two subgroups, left and right subgroups. The left subgroup is assigned to the same label of the last trusted sentence that precedes it and the right subgroup is assigned to the same label of the first trusted sentence that follows it. Figure 3.4 illustrates this criterion.

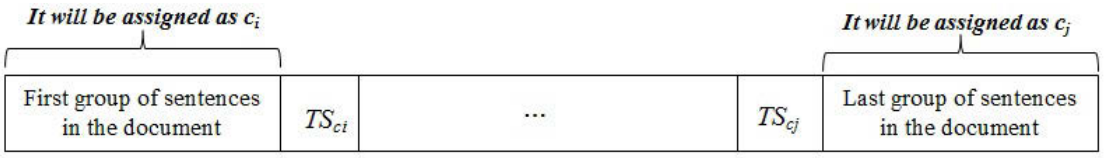

FIGURE 3.2: The illustration of criteria 2 and 3 of the proabability indication procedure. TS_{ci} and TS_{cj} are trusred sentences for classes c_i and c_j, respectively.

In the Ezekiel-Job document, by setting the value of q to be 5.0, 98.8% of the Ezekiel sentences and 99.1% of the Job sentences are correctly labeled. The overall purity result of all sentence classification is 99.0%. Table 3.4 shows the number of classified

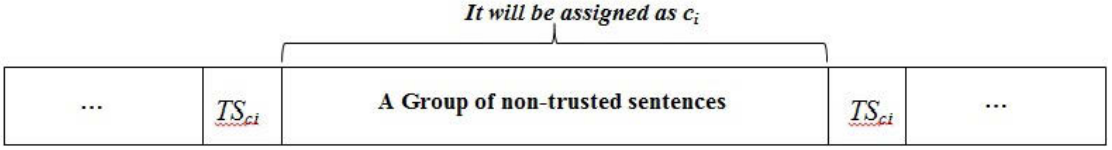

FIGURE 3.3: The illustration of criterion 4 of the proabability indication procedure. TS_{ci} is a trusred sentence for class c_i.

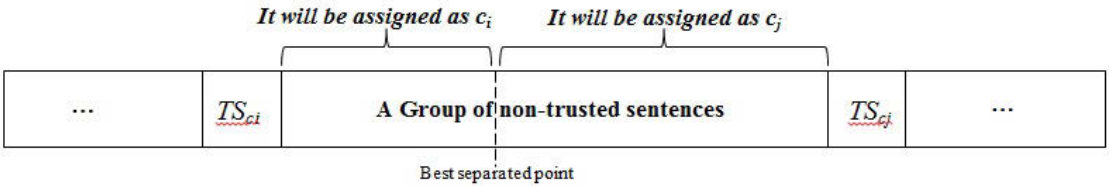

FIGURE 3.4: The illustration of criterion 5 of the proabability indication procedure. TS_{ci} and TS_{cj} are trusred sentences for classes c_i and c_j, respectively.

sentences and the number of correctly classified sentences based on the five criteria of the probability indication procedure in the Ezekiel-Job document.

Table 3.5 shows the purity results obtained by using different values of q on the Ezekiel-Job document. It is clear that our approach achieved highest purity results when we set 5.0 to q.

3.7 Experiments

In this section, the performance of the proposed approach is evaluated and compared with state-of-the-arts on three benchmark datasets widely used for authorship detection. We have used these datasets because the author of each document is known with certainty and because they are canonical datasets that have served as benchmarks for Koppel et al. (2011a), Akiva and Koppel (2012) and Akiva and Koppel (2013).

In this thesis, experimental results are evaluated by using *Purity* (Zhao and Karypis, 2001; Manning et al., 2008b; Amigó et al., 2009). The purity measure focuses on the frequency of the most common category in each class. Assuming that $L = \{L_1, L_2, \cdots, L_N\}$ is the set of classes to be evaluated, $U = \{U_1, U_2, \cdots, U_N\}$ is the set of categories, N is the number of classes (or categories) to be evaluated, and T is the number of observations, the purity is computed by taking a weighted average of maximal precision values, as shown in Eq. 3.4:

$$Purity = \sum_{i=1}^{N} \left(\frac{|L_i|}{T} \max_j P(L_i, U_j)\right) \times 100\%, \tag{3.4}$$

TABLE 3.4: The classified sentences and correctly classified sentences of the Ezekiel-Job document by applying the five criteria of the probability indication procedure

Criterion Number	**Ezekiel Class**		**Job Class**	
	Classified	**Correctly Classified**	**Classified**	**Correctly Classified**
Criterion 1	923	922	404	399
Criterion 2	0	0	0	0
Criterion 3	0	0	41	41
Criterion 4	328	328	525	519
Criterion 5	16	8	54	50
Total	**1,267**	**1,258**	**1,024**	**1,009**

TABLE 3.5: The purity results obtained by using different values of q in Criterion 1 of the probability indication procedure on the Ezekiel-Job document

	$q = 1.0$	$q = 2.0$	$q = 3.0$	$q = 4.0$	$q = 5.0$	$q = 6.0$	$q = 7.0$	$q = 8.0$	$q = 9.0$
Purity	96.4%	98.3%	98.8%	98.8%	99.0%	98.8%	98.3%	97.1%	96.8%

where P represents the precision of a class L_i for a given category U_j and is defined as:

$$P(L_i, U_j) = \frac{|L_i \cap U_j|}{|L_i|}. \tag{3.5}$$

3.7.1 Datasets

We use four datasets to test our approach and show the adaptability of our approach to different types of documents.

The first dataset, referred to as "The Becker-Posner Blog" (www.becker-posner-blog.com), is a group of 690 blogs written by the Nobel Prize winning economist Gary Becker and the legal scholar and federal judge Richard Posner, where each blog contains on average 39 sentences. The Becker-Posner Blog was started in 2004 to discuss current issues of law, economics and policy in a dialogic format. It provides a good basis for inspecting the performance of various approaches on documents where the topics among authors are not differentiated. That means, we cannot rely on the topics to help us distinguish the authors.

We test our approach on the second dataset, a group of 1,182 *New York Times* articles. These articles, having diverse topics, were written by four columnists (see Table 3.6). We use this corpus in order to evaluate the performance of the proposed approach on documents that are written by more than two authors (i.e., three or four authors).

TABLE 3.6: Statistics of the *New York Times* articles.

Columnist Name	Number of Opinion Articles	Number of Sentences
Thomas Friedman (TF)	279	11,230
Maureeen Dowd (MD)	299	11,660
Paul Krugman (PK)	331	12,634
Gail Collins (GC)	273	11,327

The third dataset tested is a group of five biblical books written by five authors (see Table 3.7). These books are related to two genres of literature, wisdom and prophetic. Note that, we adopted biblical books for three reasons. First, this corpus is highly motivated, since various researchers have been working on authorship analysis of biblical literature for centuries. Second, written in Hebrew language, this corpus gives an opportunity to test non-English documents. Third, because the five bible books are related to two literatures, it allows to evaluate the effectiveness of our approach in documents created by merging two books of the same literature.

TABLE 3.7: Statistics regarding the five Bible books.

	Author Name	Chapter Numbers	Literature Genre	Number of Sentences
1	Proverbs (Prov)	1-31	Wisdom	915
2	Jeremiah (Jer)	1-52	Prophetic	1,364
3	Ezekiel (Eze)	1-48	Prophetic	1,273
4	Isaiah (Isa)	1-35	Prophetic	676
5	Job	3-41	Wisdom	1,018

In order to show that the proposed approach can work with authentic documents, we test the proposed approach on a very early draft of a scientific paper[1] produced by two Ph.D students (Students A and B) in our research team. To use this document, we have ignored all the figures as well as all metadata (e.g., titles, author names, references and citations). The paper consists of 313 sentences and has 6 sections (including the Abstract and Conclusion). Each author has written 3 sections. Student A has written 41.9% of sentences of the paper (i.e., 131 sentences) and Student B has written 58.1% of sentences of the paper (i.e., 182 sentences).

3.7.2 Experimental Results

We conduct our experiments on four different datasets, each dataset has its characteristics which yield us to use it. In our experiments (excluding the fourth dataset), the

[1]The paper is entitled *"Cryptography-Based Secure Data Storage and Sharing Using HEVC and Public Clouds"* and is available online.

TABLE 3.8: Purity comparison on a document of Becker-Posner Blogs. Approaches compared: 1- Akiva and Koppel (2012), 2- Akiva and Koppel (2013) and 3- Our approach.

Document	1	2	3
Becker-Posner Blogs	94.0%	94.9%	96.6%

merged documents are created in the same way as we have discussed before (i.e., Section 3.2). We use the same values of the parameters as we have used in the Ezekiel-Job document. We use our proposed approach in these datasets where the objective is to decompose the multi-author document into N groups according to the authorship.

3.7.2.1 Results on Becker-Posner Blogs Dataset (Controlling for Topic)

In the first dataset, each author has written for a lot of different topics, and there have been some topics taken by both authors. Therefore, there is no topic indication to distinguish between the two authors. We have achieved a purity result equal to of 96.6% when testing on this dataset. This result is gratifying in this merged document that has more than 246 transitions between sentences written by the two authors and more than 26,900 sentences.

In Table 3.8, we show the comparison between our approach and the approaches in Akiva and Koppel (2012) and Akiva and Koppel (2013). As shown in Table 3.8, the purity result of our approach is higher than that of the other two approaches.

3.7.2.2 Results on *New York Times* Articles Dataset (N $\geq$ 2)

This dataset contains articles written by four authors. First, we test our approach using the merged documents created by any pair of the four authors. Table 3.9 shows the purity results of the six documents created by merging any pair of the four authors and the number of turns between authors in each document.

TABLE 3.9: The purity results of documents created by merging any pair of the four *New York Times* columnists using our proposed approach.

	Document	**No. of Turns**	**Our Proposed Approach**
1	TF-PK	251	95.6%
2	GC-PK	253	93.7%
3	GC-TF	242	96.1%
4	MD-PK	255	95.5%
5	MD-TF	249	93.3%
6	MD-GC	251	93.8%

As shown in Table 3.9, the results are noticeable and range from 93.3% to 96.1%. For comparison, the result can be as low as 88.0% when applying the approach in Akiva and Koppel (2013) on some of the merged documents.

To prove that our approach can also work well with merged documents written by more than two authors, we have created merged documents written by any three of these four authors and formed four merged documents. We have also created a merged document written by all four *New York Times* authors. Then, we apply our approach on these documents. In Figure 3.5, we show the purity results of our approach for segmentation on these documents. It is obvious that our approach achieves high purity results even when the documents are written by more than two authors. Furthermore, Figure 3.5 compares our results with the results achieved by Akiva and Koppel (2012) and Akiva and Koppel (2013). It shows that our approach has given consistent results and better performance than the approaches of Akiva and Koppel (2012) and Akiva and Koppel (2013).

3.7.2.3 Results on the Biblical Books Dataset

In these experiments, we use two literature types of biblical books. We create merged documents written by any pair of authors. The resulted documents may belong to either the same literatures or different literatures (see Table 3.7).

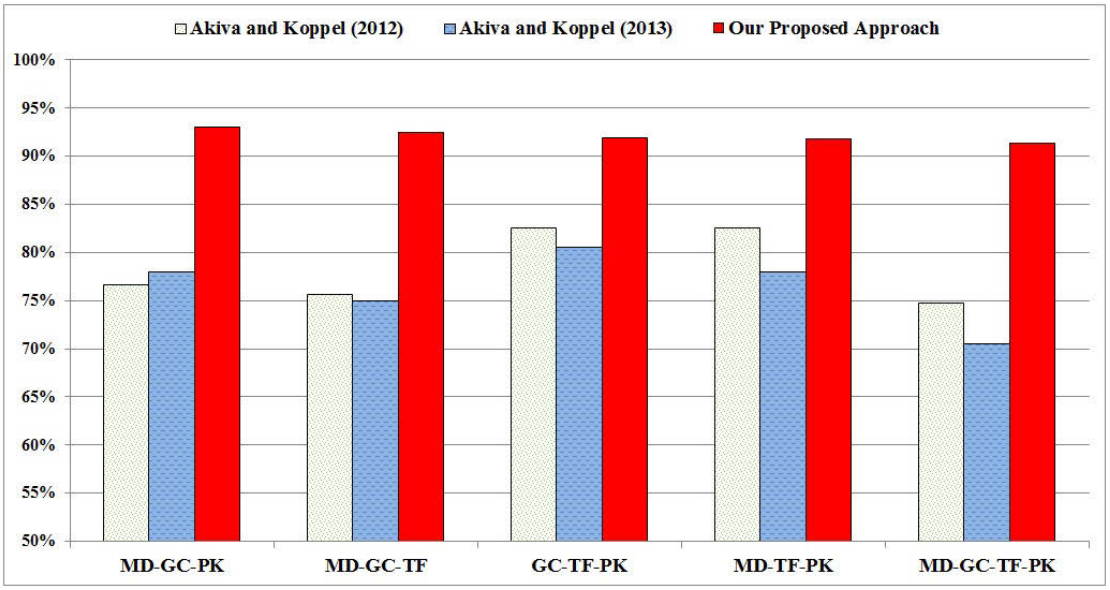

FIGURE 3.5: Purity results of the approaches proposed by Akiva and Koppel (2012), Akiva and Koppel (2013) and our proposed approach using documents created by three or four *New York Times* authors.

TABLE 3.10: Purity comparison on documents composed by merging two biblical books of *different literatures*. Approaches in comparison: 1- Koppel et al. (2011a), 2- Akiva and Koppel (2013), 3- Akiva and Koppel (2013)- SynonymSet, 4- Our Proposed Approach.

Document	1	2	3	4
Jer-Prov	72.7%	97.0%	75.0%	99.0%
Isa-Job	82.2%	98.7%	89.1%	98.7%
Eze-Prov	76.6%	98.7%	90.8%	97.9%
Isa-Prov	70.4%	95.0%	85.0%	97.9%
Eze-Job	85.9%	98.7%	95.0%	99.0%
Jer-Job	87.3%	98.0%	93.1%	97.8%
Overall	***79.2%***	***97.7%***	***88.0%***	***98.4%***

In Tables 3.10 and 3.11, we show the purity results achieved by applying the proposed approach on the documents created by merging two biblical books of different literatures (Table 3.10) and the documents created by merging two biblical books of the same literature (Table 3.11), respectively. In both cases, the purity results of the proposed approach are compared with the approaches of Koppel et al. (2011a), Akiva and Koppel (2013) and Akiva and Koppel (2013)-SynonymSet. The purity results of the approach of Akiva and Koppel (2012) are used further for comparison in Table 3.11.

TABLE 3.11: Purity comparison on documents composed by merging two biblical books of the *same genre*. Approaches in comparison: 1- Koppel et al. (2011a), 2- Akiva and Koppel (2012), 3- Akiva and Koppel (2013), 4- Akiva and Koppel (2013)-SynonymSet, 5- Our Proposed Approach.

Document	**1**	**2**	**3**	**4**	**5**
Job-Prov	84.5%	84.9%	93.9%	82.0%	95.2%
Jer-Eze	82.0%	87.6%	96.6%	95.9%	97.0%
Isa-Jer	71.8%	63.4%	66.7%	82.7%	71.0%
Isa-Eze	78.9%	76.0%	80.0%	88.0%	82.7%
Overall	***79.3%***	***78.0%***	***84.3%***	***87.2%***	***86.5%***

As shown in Table 3.10, the results using our proposed approach in the documents created by merging two bible documents of different literatures are interesting with purity results ranging from 97.8% to 99.0%. Table 3.10 also shows that the overall purity result of our proposed approach is remarkably better than those obtained using the approaches of Koppel et al. (2011a), Akiva and Koppel (2013) and Akiva and Koppel (2013)-SynonymSet. In Table 3.11, it is clear that the purity results obtained with our approach in the documents created by merging two bible documents of the same are encouraging. Furthermore, Table 3.11 shows that the purity results achieved by our approach is higher than those obtained using the approach of Koppel et al. (2011a), Akiva and Koppel (2012) and Akiva and Koppel (2013), and less than that obtained using the approach of Akiva and Koppel (2013)-SynonymSet. Note that, two of them, i.e., Koppel et al. (2011a) and Akiva and Koppel (2013)-SynonymSet, are specially developed for biblical books only, and not applicable for other documents.

3.7.2.4 Results on Authentic Document

In order to examine the performance of the proposed approach on non-artificial documents, a scientific paper initially written by two PhD students (Students A and B) is used. We apply our approach on the scientific paper to decompose the 313 sentences of the paper into two authorial components. Our proposed approach obtains a purity of 92.0% on the scientific paper. It is noticeable that the purity result obtained using the proposed approach on a non-artificial, scientific paper is very promising.

3.8 Summary

This chapter has proposed an unsupervised method for decomposing a multi-author document by authorship. It is assumed that no information about the document and the authors is available. The segment elicitation procedure, which aims to exploit the differences in the posterior probabilities of the Naive Bayesian model, is applied in order to select vital segments of each cluster. These selected vital segments are used to train a more effective classifier. The probability indication procedure is proposed in order to enhance the purity of sentence classification process. The procedure consists of five criteria. It works by selecting trusted sentences from the document and using them to re-classify each sentence of the document into the author's class. The approach has been tested using four datasets, of which every one has its own characteristics. It is clear that the approach has achieved significantly high purity results in these datasets, even when there is no topic indication to differentiate sentences between authors, and when the number of authors exceeds two. Our results tested on these datasets have shown significantly better than those using the approaches in Koppel et al. (2011a), Akiva and Koppel (2012) and Akiva and Koppel (2013). Furthermore, the approach can also compete with the approach proposed in Akiva and Koppel (2013)-Synonyms, which is only valid for Bible documents.

In the next chapter, the proposed approach presented in this chapter will be further extended and an unsupervised, hierarchical Naive Bayesian framework for authorship-based multi-author document segmentation will be presented.

For brevity, in the following chapters of this thesis, we denote the proposed approach presented in this chapter by *Proposed-1*.

Chapter 4

An Unsupervised Hierarchical Framework for Authorship-based Segmentation of a Multi-Author Document

Segmenting a document collaboratively written by multiple authors into distinct authorial components plays an increasingly important role in many applications and has great significance on security and forensic investigation. Despite its value, there is very little work reported. Existing approaches have limitations on topics, specific languages, styles of writing, or requiring the availability of the profiles of authors. In this chapter, we formulate our proposed unsupervised Naive-Bayesian-based approach (i.e., *Proposed-1*), which is presented in Chapter 3, into a general unsupervised, hierarchical learning framework. During the first level of learning, we first look at a consecutive number of sentences (i.e., segments) and cluster them based on the writing styles reflected by the group of sentences. The clustering results are used in a probability-based segment elicitation procedure to select the most discriminative segments (i.e., significant segments), which are then employed to train the first stage classifier. Based on the initial classification results, we further create a new, more accurate training dataset and perform a second level of learning based on selected trusted sentences. The key novelty and benefit

of the two-level hierarchical learning framework lies in two main aspects: 1) We start from estimating the writing styles reflected by segments and produce the initial class information of data; 2) Then, we take advantage of the difference in the posterior probabilities of the Naive-Bayesian model and create meaningful training datasets with a high precision for the use of accurate supervised learning. We evaluate the performance of the proposed approach on three benchmark datasets widely used for authorship analysis. A scientific paper is also used to demonstrate the performance of the approach on authentic documents. Experimental results show the superior performance of the proposed approach over the state-of-the-arts.

4.1 Introduction

With the wide availability of online communication facilities, documents collaboratively written by multiple authors are widely deployed in the internet. Co-authored documents can be found in books, academic papers, academic thesis and blog posts. Segmenting a multi-author document into distinct authorial components thus plays an increasingly important role in many applications, such as for forensic investigation (Abbasi and Chen, 2005; Grant, 2007), plagiarism detection (Zu Eissen et al., 2007), civil law (i.e., disputed copyright issues) and commercial purpose (e.g., defining authors' contributions in multi-author documents).

Many approaches have been reported to handle various problems related to document segmentation. Some of these approaches are based on segmenting the document according to topics instead of authors. For example, the works reported in Chen et al. (2009), Rosen-Zvi et al. (2010), Joty et al. (2013) and Savoy (2013a) have applied the probability topic model Hofmann (1999) for topic-based segmentation. The work of Han et al. (2014) has considered the text segmentation according to author's location. It has applied a number of feature selection methods to identify location indicative words. The works described in Koppel et al. (2006), Iqbal et al. (2013), Schaalje et al. (2013) and Segarra et al. (2014) are for identifying the author of a document written by only a single author. It was assumed that those single authors were known. In order to assign a document to its author, supervised learning approaches were utilised to learn the distinctive profiles of

individual authors, based on labelled training data. Accordingly, the approaches applied various classifiers to determine the most likely or most unlikely author given an anonymous document. The classifiers that can be applied include Naive-Bayesian model (Peng et al., 2004; Savoy, 2013a), Support Vector Machine (SVM) (Abbasi and Chen, 2008; Stamatatos, 2008) and decision trees (Zheng et al., 2006; Koppel et al., 2009b). In many cases, a distance measure, such as Delta rule (Jockers and Witten, 2010; Savoy, 2013a) and Chi-square distance (Grieve, 2007; Luyckx and Daelemans, 2008; Savoy, 2013b), was defined and employed as a similarity measurement to determine the author of an anonymous document. The most likely author of a document is the one that corresponds to the smallest distance. Later, semi-supervised (Qian et al., 2014) and unsupervised (Layton et al., 2013) approaches were proposed to cluster single-authored documents according to authorship with unlabelled data and had achieved good results. The work of Daks and Clark (2016) has proposed an unsupervised technique for clustering a collection of documents according to their authorships. However, they have supposed that each document has been written by only one author.

However, for segmenting a document written by multiple authors according to the authorship, there is very little work reported. In Graham et al. (2005), the authors presented a supervised method for segmenting a document into authorial components assuming that each paragraph in the document was written by one single author. The work in Koppel et al. (2011a) considered the segmentation of a multi-authorship document, where each paragraph was not necessarily written by a single author. However, this approach dealt with documents in Hebrew language only and thus could only be applied on particular types of documents such as Bible books written in Hebrew. Furthermore, this method required the concordance between synonyms. Another limitation of the approach is that it deals with only the documents artificially combined from two Bible books written by two authors respectively. These shortcomings were mitigated in Akiva and Koppel (2012) and a new unsupervised method was proposed relying on a distance measurement in a clustering and classification process. However, the resultant accuracy was not satisfactory. Later, Akiva and Koppel (2013) further extended their work in Akiva and Koppel (2012) and developed a generic unsupervised method. The performance of this method degraded rapidly when the number of authors increased.

In Giannella (2015), the author addressed the problem of unsupervised decomposition of a multi-author document, but the approach had only been tested on documents where the number of switches between the authors was very small. Moreover, the performance of the approach was very sensitive to its parameter setting.

To address the above-mentioned limitations, in this chapter, we aim to develop an unsupervised approach for separating out distinct authorial components of a multi-author document based on authorship. This approach should be able to handle documents of any types and any topics written by any number of authors, requiring no extra information about the document's context or authors' writing profiles. Precisely, suppose there are N authors who have collaborated in creating a document. It is assumed that each sentence is completely written by only one of the N authors. Our objective is to segment the document into N authorial components, so that sentences that are written by one author are grouped in one component.

To solve this problem, in Chapter 3, we have proposed a Naive-Bayesian-based approach for segmenting a multi-author document according to authorship. The approach, which is completely unsupervised, has been demonstrated to be able to differentiate sentences (instead of paragraphs) in a document of any types and any topics according to authorship, even when the number of authors increase. Following this direction, in this chapter, we further develop the idea and formulate it into an unsupervised, hierarchical Naive Bayesian framework for multi-author document segmentation. The two procedures, namely the segment elicitation procedure and probability indication procedure, originally proposed in Chapter 3 for the one-level approach are now creatively modified and used in two levels respectively in this chapter.

As a highlight, the new contributions of this chapter are as follows.

- We formulate our previously proposed unsupervised Naive-Bayesian-based approach, which is presented in Chapter 3, into a more general unsupervised, two-level hierarchical learning framework. In this two-level learning framework, the purpose of the first level of learning is to generate a discriminative training dataset

from the unlabeled input data using a modified probability-based segment elicitation procedure, and use it to train the first-stage classifier. The results of the first-stage classifier are utilised to generate a new but more accurate training dataset, which is then used for training the second-stage classifier in order to achieve better purity results.

- Under this framework, in this chapter, we propose a second level learning, which is based on the initial classification results of the first level. We further extend our idea of the modified probability-based segment elicitation procedure and identify the most trusted sentences to create a more accurate and discriminative dataset of sentences for the second level learning. Then, a modified probability indication procedure is applied to refine the outcomes.

- Different from our approach presented in Chapter 3, the proposed approach employs the Bernoulli distribution to effectively model the conditional probabilities of words of a document.

- Last but not the least, more comprehensive experiments are conducted to demonstrate the superior performance of our approach on both artificial benchmark datasets and authentic scientific documents. When tested on single-topic documents, the proposed approach has also achieved very promising results, showing its independence on the topic of the document.

The rest of the chapter is organised as follows. Section 4.2 gives the overview of the proposed framework. This is followed by Sections 4.3 and 4.4 describing the two levels of learning respectively. Finally, experimental results are presented in Section 4.5. Finally, Section 4.6 presents the summary of the chapter.

4.2 Framework of the Proposed Approach

Figure 4.1 shows the proposed framework.

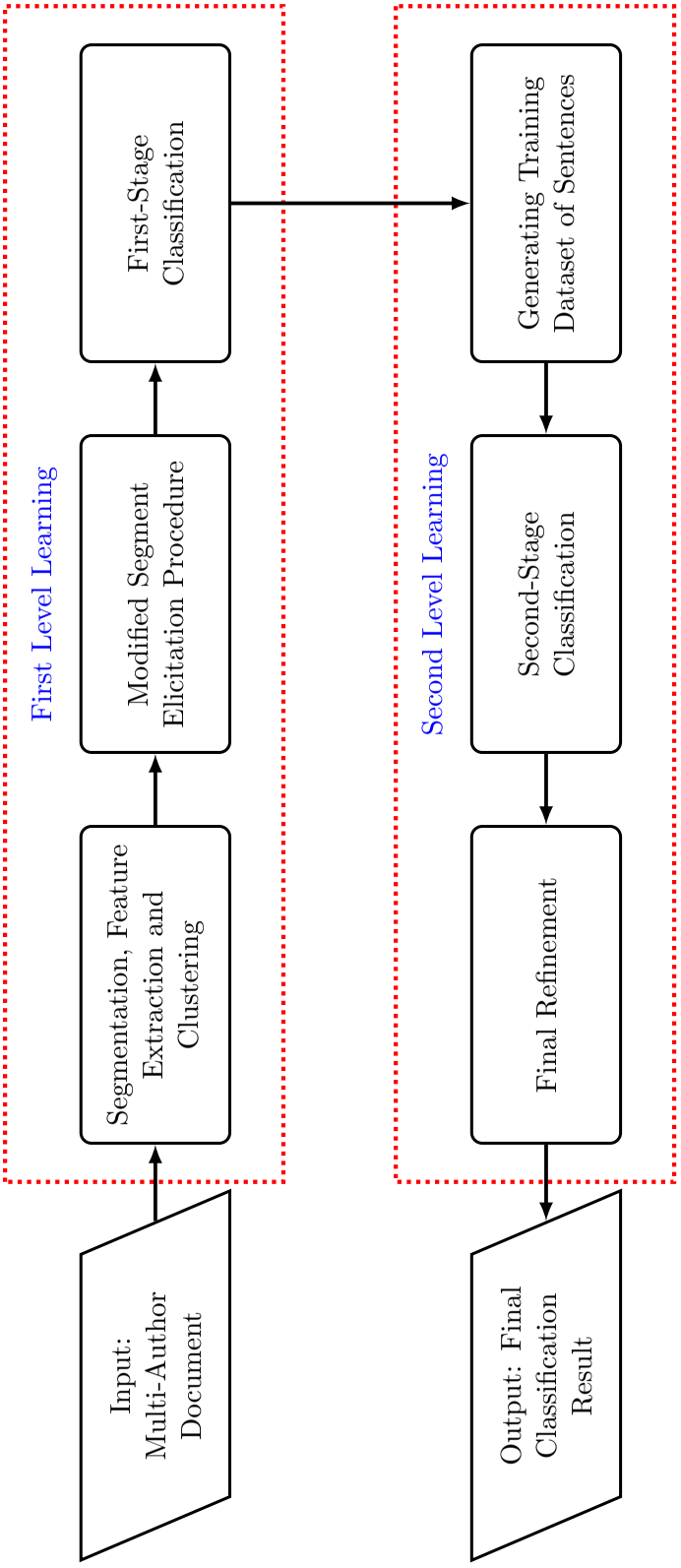

FIGURE 4.1: The proposed two-level, unsupervised learning framework.

As shown in Figure 4.1, in order to estimate the writing styles of authors from unlabelled input data, we first divide a multi-author document into segments and extract features vectors from these segments. The vectors are then clustered into N clusters aiming at a high recall rate for each cluster. With the clustering results, the most significant vectors from each cluster that can represent authors' writing styles are then selected, referred to as *significant segments*, using a modified probability-based segment elicitation procedure. These significant segments, bearing the labels of their corresponding clusters, are then used to train the first-stage classifier. This initial first-stage classifier is then applied to classify all sentences in the document, so the first level, namely unsupervised learning, concludes. During the second level of learning, with the classification results of the first level learning, a procedure based on posterior probability of each sentence' classification score is applied to generate a more accurate training dataset to train the second-stage classifier. In the end, a re-classification process is performed on all sentences in the document using the second-stage classifier. Finally, using the results of the second classification, a modified probability-based refinement procedure is applied to further improve the accuracy performance of the classification.

In the following two sections, we illustrate our proposed approach based on the two levels of learning described above.

4.3 First Level Learning

4.3.1 Segmentation, Feature Extraction and Clustering

In order to carry out clustering based on authors' writing styles, we first divide the merged document into *segments*, each containing v consecutive sentences from the document. The number of sentences in each segment (i.e., v) is estimated according to the number of sentences in the document. We set 30 and 10 to v for a long document (contains more than or equal to 500 sentences) and a short document (contains fewer than 500 sentences) respectively. Note that, the length of the segment, v, is selected in a way that the decomposition of the document can produce segments with a sufficient length

to reflect the authors' writing styles, and also, it should provide an adequate number of segments (so as to be used later to form a training dataset in the first level learning).

Each of these segments is then represented in a vector formed in the way shown below so as to be clustered into components, each representing a different author's writing style. We use all words appearing at least three times in the document as features. We use a binary vector, of which each bit represents whether an individual feature (i.e., a word in the feature set) does or does not occur in the segment. Note that, the binary nature of the feature set is critical and important. The advantages of using a binary vector representation has been seen in Koppel et al. (2011a) from the fact that the use of frequencies of common words (instead of a binary vector) to represent a segment fails completely for clustering segments according to authorship.

Then, we apply the Gaussian Mixture Models (GMMs) to cluster the feature vectors to N multivariate Gaussian densities. Our experiments have shown that the GMMs clustering technique is a powerful and robust technique even when the number of the models increases. The GMMs clustering technique is discussed in Section 2.4.

4.3.2 Modified Segment Elicitation Procedure

With the initial clustering results, we can assign labels to the segments in each cluster to bear the labels of their cluster. These segments can then be used to produce a dataset for supervised training.

However, there are some segments mistakenly-assigned to a wrong cluster during the clustering process. These mistakenly clustered segments only confuse the writing styles reflected by other segments in the cluster, so they should be removed. Ideally, only the correctly clustered segments in each cluster are to be retained and selected to form the training dataset. Moreover, the segments that reflect the writing styles of more than one author may lead to confusion in differentiating the writing style of different clusters and must be removed as well. Thus, we propose to identify the most representative segments in terms of authors' writing styles and then form a labelled training dataset.

We first represent all segments in all clusters as binary vectors using a feature set containing some most frequently-appearing words in the document. We have experimentally compared the impact of the number of the frequently-appearing words on the purity results, and found choosing 1500 most frequently-appearing words performed best on all our tested documents. Then, segments in each cluster are filtered, and only the segments that can best represent the writing style of the cluster are chosen. The writing style of a specific cluster represents the writing style of a specific author. We call these segments *significant segments*.

Let us again denote each segment as x, where x is an D-dimension binary vector, i.e., $x = \{x^1, x^2, \cdots, x^D\}$, with D equal to 1500 here. For simplicity, we use 'segment' to refer the feature vector of the segment when there is no ambiguity. Henceforth, we deem all of the segments as labelled, based on the results of the clustering obtained in Section 4.3.1. The segments that are assigned into one cluster are considered as segments having the same label different from the labels assigned to other clusters. Therefore, each segment x is considered to be classified into one of the N classes, denoted by $\boldsymbol{C} = \{c_1, c_2, \cdots, c_N\}$.

For each segment x in a class c_i, $i \in \{1, 2, \cdots, N\}$, the posterior probability of a class for the given segment, $p(c_i|x)$, is computed using the Naive-Bayesian theorem as:

$$p(c_i|x) = \frac{p(c_i)p(x|c_i)}{p(x)}, \qquad i \in 1, 2, \cdots, N. \tag{4.1}$$

More details about the Naive-Bayesian model can be found in Section 2.2.

Different from the segment elicitation procedure presented in Chapter 3, where the non-occurring words within each segment are completely ignored for estimating a conditional probability of the segment (i.e., $p(x|c_i)$), in this chapter, the Bernoulli distribution (Evans et al., 2000) is used in order to estimate the segment conditional probability where the presence and absence of words within the segment are taken into account. The conditional probability of the segment is shown in Eq. 4.2.

$$p(x|c_i) = \prod_{d}^{D} \theta_{di}^{x^d} (1 - \theta_{di})^{1-x^d}, \tag{4.2}$$

where θ_{di} is a probability of class c_i to generate the feature (i.e., word) x^d. The θ_{di} is estimated as follows.

$$\theta_{di} = \frac{n_{di}}{n_i} , \tag{4.3}$$

where n_{di} is the number of times that feature x^d (i.e., word d) occurs in a segments of class i and n_i is the number of segments of class i.

To find the significant segments of each class, we consider the differences in the posterior probabilities of each segment according to the other classes. Expressly, for each segment in a class, we compute the differences between the posterior probability of that segments in its class and the maximum posterior probability of that segment in other classes. Then, we select $s\%$ of them which have the biggest differences as significant segments of that class. We follow our proposed approach presented in Chapter 3 (i.e., *Proposed-1*) and assign 80 to s.

4.3.3 First-Stage Classification

The selected significant segments, now with labels, of N classes are then used to train the first stage classifier so as to classify each sentence in a document into one of the N classes.

In order to capture the differences among the writing styles reflected by sentences (rather than segments), which may contain just a few words, a new feature set is created, which is based on all words (counted once only if they appear twice or more) that appear in the document. The list of words is used for representing all of the significant segments of all classes as binary vectors. The multivariate Bernoulli Naive-Bayesian model is then used to construct a classifier.

Let us assume that the merged document consists of T sentences. The classifier is used to predict the class label c_i, $i \in \{1, 2, \cdots, N\}$ of sentence s_t, where $t = \{1,2, \cdots, T\}$, using Eq. (4.4) as:

$$\hat{c}^t \leftarrow \underset{c_i}{\text{argmax}}\ p(c_i|s_t), \qquad i = 1, 2, \cdots, N, \tag{4.4}$$

where $\hat{c}^t$ is the predicted class label of sentence s_t.

The computation of $p(c_i|s_t)$ is similar to the computation of the posterior probability shown in Eq. (4.1), with the only difference that the segment x in Eq. (4.1) is replaced by the sentence s_t in Eq. (4.4).

4.4 Second Level Learning

The above classification results are obtained based on the initial estimation of labels from clustering groups (i.e., segments) of consecutive sentences. A more powerful training dataset with more accurate labels can further improve the classifier's robustness and performance. Therefore, with the availability of the high-quality first-stage classifier, the classification performance can be further improved through a second level supervised learning.

In this level, we apply a similar idea to create a strengthened dataset for training using the classification results obtained from the first level learning. Then, the new training dataset is used to train a new supervised classifier to re-classify each sentence of the document to a correct author's class (i.e., second-stage classification).

4.4.1 Generating Training Dataset of Sentences

Different from the procedure of generating supervised training data in the first level, where the top 80% most significant segments from each cluster are selected, in this level, *trusted sentences* are selected based on their posterior probabilities to produce a strengthened training dataset of sentences. The sentence conditional probability that is used in estimating the posterior probability of a sentence is also computed by using the Bernoulli distribution. The new training dataset is produced according to the following criteria.

1. Any sentence in the document is deemed as a *trusted sentence* if its posterior probability of its class is greater than its posterior probabilities of all other classes by more than a threshold q_1. Each trusted sentence remains in its class and is added into the new training dataset.

2. If a sentence is the first sentence classified as a trusted sentence in a document, then all sentences before this sentence are grouped to the same class as the trusted sentence. All these sentences with the same class label as that of the trusted sentence are added into the new training dataset.

3. If a sentence is the last sentence classified as a trusted sentence in a document, then all sentences after this sentence are grouped to the same class as that of the trusted sentence. All these sentences with the same class label as the trusted sentence are added into the new training dataset.

4. If two consecutive trusted sentences belong to the same class, then all sentences between these two sentences are grouped to the same class of the two trusted sentences and added into the new training dataset.

From the above, we can see that the new training dataset has the following features.

1. The new training dataset contains only the sentences that have a high chance to possess correct labels. Therefore, the percentage of correctly labelled sentences in this dataset is high. Our experiments have shown that the precisions of the correctly labeled dataset in the new training dataset are better than in the previous training dataset used in the first-stage classification. More details can be found in the Experiments section.

2. Each feature vector in the new training dataset represents one sentence. By contrast, each vector in the training dataset used for the first level learning represents one segment consisting of v (i.e., 10 or 30) sentences. Therefore, the number of sentences in the new training dataset is much larger than the number of segments used in the training dataset for the first level learning.

The new training dataset is used in the second-stage classification in the next subsection to reclassify each sentence in the document to an author's class.

4.4.2 Second-Stage Classification

The new training dataset, created as above, containing more data of higher precision, is more accurate and can enhance the performance of the training. Therefore, a second level supervised learning is applied using the new training data. The goal is to learn a more accurate classifier using the strengthened training data to reclassify each sentence of the merged document into one of the N classes. Again, a new, second-stage Naive-Bayesian classifier is learned. The same as before, the newly learned classifier is used to predict the class label c_i, $i \in \{1, 2, \cdots, N\}$, of all sentences in the document using Eq. (4.4).

4.4.3 Final Refinement

Our experiments show that the performance of the second level supervised learning is much better than that of the first level unsupervised learning. However, its performance can still be further improved.

In Chapter 3, a refinement procedure to further improve the purity results of the classifier has been proposed. This refinement is based on a probability indication procedure. It selects the most significant and trustful sentences from the classification results and then uses them to make adjustments and assign each of the potentially misclassified sentences into one of the N classes.

Following this idea, in this chapter, a modified version of the procedure is proposed to enhance the purity results from the following aspect: Unlike the original procedure, the modified version of the procedure takes into account the presence and absence of all words within the sentence in computing the conditional probabilities of words.

The modified probability indication procedure is based on five criteria. The first four criteria are the same as mentioned in Sub-section 4.4.1 except that the threshold used

in Criterion 1 is replaced by a different threshold q_2. A fifth criterion is added, which is stated as follows.

5. If two consecutive trusted sentences belong to different classes, then the sentences between these two trusted sentences are divided into two parts, at the point where the posterior probabilities of both parts reach to maximum. All sentences on the left or right part are assigned to the same class as the trusted sentence on the left or right, respectively.

4.5 Experiments

To demonstrate the performance of our proposed framework, we test our approach and compare it with the state-of-the-art approaches on three benchmark datasets widely used for authorship analysis. Furthermore, since all of the three benchmark datasets are artificially documents, we also test the proposed approach on an authentic scientific paper.

4.5.1 Datasets

The three benchmark datasets used for experiments in Chapter 3, i.e., Becker-Posner blogs, *New York Times* articles and bible books, are used in this chapter. Furthermore, the scientific paper that is used in Chapter 3 is also employed in this chapter.

Regarding the first dataset, i.e., Becker-Posner blogs, the work in Giannella (2015) manually created six single-topic documents from the Becker-Posner blogs in order to evaluate the performance of his work (see Table 4.1), where each document has sentences representing only one single topic. In this chapter, we use these documents because each of these documents has only one single topic, all these documents are short and the total number of consecutive sentences of each author in these documents is relatively small. Therefore, this corpus makes the task of distinguishing the sentences in a document, according to authorship, rather than topics, be more challenging.

TABLE 4.1: Statistics of the six single-topic documents created from the Becker-Posner Blogs.

Topics	Author order and number of sentences per author
Traffic Congestion (TC)	Becker(57), Posner(33), Becker(20)
Senate Filibuster (SF)	Posner(39), Becker(26), Posner(28), Becker(24)
Microfinance (Mic)	Posner(51), Becker(37), Posner(44), Becker(33)
Tort Reform (TR)	Posner(29), Becker(31), Posner(24)
Profiling (Pro)	Becker(35), Posner(19), Becker(21)
Tenure (Ten)	Posner(73), Becker(36), Posner(33), Becker(19)

4.5.2 Experimental Results

We evaluate the performance of the proposed approach through a set of experiments on different documents. In the experiments regarding the three benchmark datasets, excluding the six single-topic documents on Becker-Posner blogs, we create artificially merged documents. These documents are created by using the same method that has been used in Chapter 3 (i.e., Section 3.2). We empirically assign 5.0 to the thresholds q_1 and q_2 in the modified probability-based procedure of generating training dataset of sentences and modified probability indication procedure, respectively.

4.5.2.1 Results on the Becker-Posner Blogs Dataset (*Controlling for Topic*)

In our first set of experiments, we use 690 blogs written by Becker and Posner to form merged documents. We apply our experiments regarding this dataset using two types of documents. The first type contains only a single document that has been used in the approaches of Akiva and Koppel (2012), Akiva and Koppel (2013) and *Proposed-1*, and it is the merged document containing all of the 690 blogs of both authors (i.e., Becker and Posner) and is created by using the procedure of merging documents. The resultant merged document has 26,922 sentences in total and there are hence 246 turns from Becker's sentences to Posner's sentences and from Posner's to Becker's. The document covers a lot of different topics, and some of these topics are shared by both authors. Therefore, the topics are not differentiated according to the authorship. The second

TABLE 4.2: Purity results of the document of all Becker-Posner blogs using the approaches of [1] Akiva and Koppel (2012), [2] Akiva and Koppel (2013), [3] *Proposed-1*, [4] First level learning and our approach.

Approach	**[1]**	**[2]**	**[3]**	**[4]**	**Ours**
All Becker-Posner Blogs	94.0%	94.9%	96.6%	96.7%	96.8%

type, which has been used in the approach of Giannella (2015), has six single-topic documents (see Table 4.1).

Table 4.2 presents the purity results on the first type of document using our approach and the approaches in Akiva and Koppel (2012), Akiva and Koppel (2013) and *Proposed-1*. As shown in Table 4.2, the purity result of our approach is noticeably higher.

In fact, it would be interesting to know the effectiveness of applying only the first level learning of our approach. Table 4.2 also presents the purity result achieved by applying the first level learning of our approach on the first type of Becker-Posner document. It is clear that the purity result achieved at this level is good and also surpasses the purity results of the other three approaches. Furthermore, the significant of two-level hierarchical learning can be observed by notifying the improvement in the purity results in the table.

Figure 4.2 compares the purity results on the second type of documents using our approach and the approach in Giannella (2015). It can be seen from Figure 4.2 that, the purity results of our approach has exceeded those of the another approach.

4.5.2.2 Results on *New York Times* Articles Dataset (N $\geq$ 2)

In this set of experiments, we use a dataset containing of 1,182 *New York Times* articles written by four authors. First, we apply our approach by using a merged document composed of the documents written by any two of the four authors. This produces six merged documents. Figure 4.3 shows the purity results of these six merged documents using *Proposed-1* and our approach.

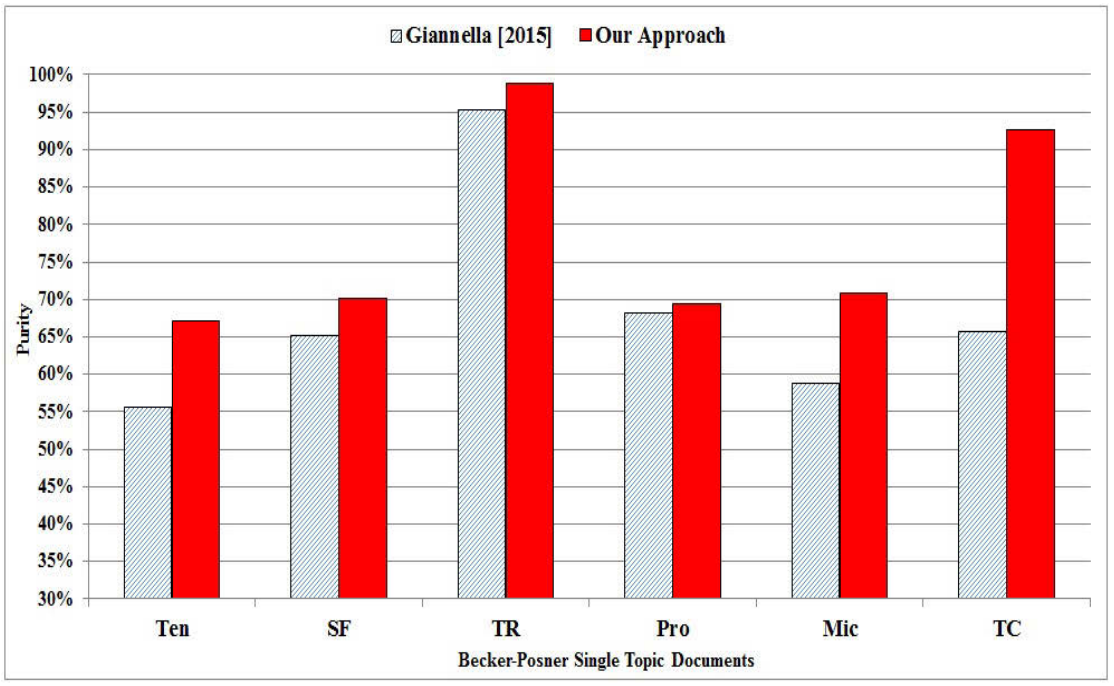

FIGURE 4.2: Comparison of the purity results obtained using our approach and the approach in Giannella (2015) on the six single-topic documents.

As shown in Figure 4.3, the results are promising. The purity results of the proposed approach on the merged documents are in the range from 94.0% to 97.0% and they exceed those obtained by our previously proposed approach (i.e., *Proposed-1*) in all of the six documents. Furthermore, as shown in Akiva and Koppel (2012) and Akiva and Koppel (2013), their results on some of the documents merged in the same way as ours show the purity results to be as low as 88.0%, which is a lot lower than our minimum result of 94.0%

The main objective of using this corpus is to examine our approach with a merged document written by more than two authors. Therefore, we create merged documents written by any three of the four columnists of *New York Times* articles. We get four merged documents. Each document has a number of sentences between 34,217 and 35,621 and more than 350 turns between the authors. Furthermore, we create a merged document written by all four authors. The merged document has 46,851 sentences and more than 500 turns between the authors. Then, we apply our approach on these documents (i.e.,

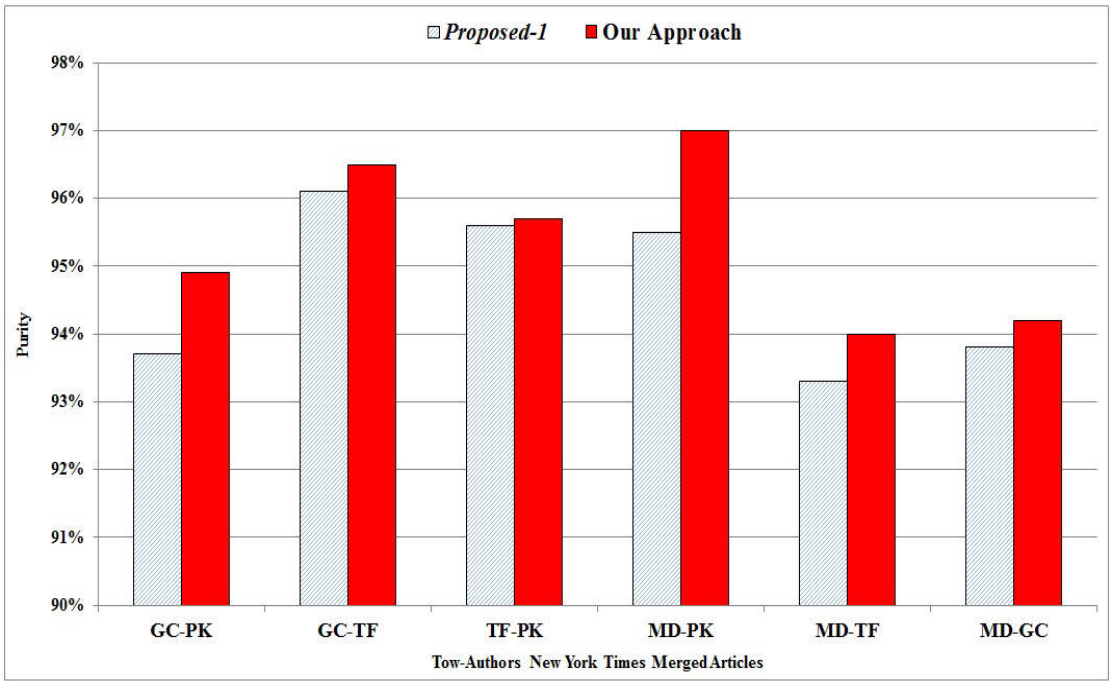

FIGURE 4.3: Comparison of the purity results obtained using the *Proposed-1* approach and our approach on the six documents created by merging *New York Times* articles of two columnists.

TABLE 4.3: Purity results of the documents merged from the articles written by three or four of the *New York Times* columnists, respectively, using the approaches of [1] Akiva and Koppel (2012), [2] Akiva and Koppel (2013), [3] *Proposed-1* and our approach.

Document	**[1]**	**[2]**	**[3]**	**Ours**
MD-GC-PK	76.7%	78.0%	93.0%	93.4%
MD-GC-TF	75.6%	75.0%	92.4%	93.0%
GC-TF-PK	82.6%	80.5%	91.9%	92.2%
MD-TF-PK	82.6%	78.0%	91.7%	92.3%
MD-GC-TF-PK	74.8%	70.5%	91.3%	91.7%

four merged documents written by three authors and one merged document written by four authors).

Table 4.3 represents the purity results of these documents using our approach and the approaches of Akiva and Koppel (2012), Akiva and Koppel (2013) and *Proposed-1*. From this table, one can find that our approach always gives highly significant results no matter if a document is written by three or four authors. Furthermore, it is seen that the results of our approach on these documents always outperform the results of the aforementioned approaches.

TABLE 4.4: Purity results of documents created by merging two bibles of different literatures. Approaches in comparison: [1] Koppel et al. (2011a), [2] Akiva and Koppel (2013), [3] Akiva and Koppel (2013)-SynonymSet, [4] *Proposed-1* and our approach.

Doc.	**[1]**	**[2]**	**[3]**	**[4]**	**Ours**
Jer-Prov	72.7%	97.0%	75.0%	99.0%	99.5%
Isa-Job	82.2%	98.7%	89.1%	98.7%	99.1%
Eze-Prov	76.6%	98.7%	90.8%	97.9%	99.0%
Isa-Prov	70.4%	95.0%	85.0%	97.9%	98.6%
Eze-Job	85.9%	98.7%	95.0%	99.0%	99.7%
Jer-Job	87.3%	98.0%	93.1%	97.8%	98.8%
Overall	***79.2%***	***97.7%***	***88.0%***	***98.4%***	***99.1%***

Results on the Bible Books Dataset

In this set of experiments, we use a dataset consisting of five biblical books related to two literatures, namely, the prophetic literature and the wisdom literature, where each book is written by an author. In order to examine our approach using these books, we create documents merged from any two of the books. Each merged document is created by using a pair of Bible books of the same literature or of different literatures.

Tables 4.4 and 4.5 show the purity results of our approach on the documents created by merging two bibles of different literatures (Table 4.4) and the documents created by merging two bibles of the same literature (Table 4.5). In both cases, the purity results of the proposed approach are compared with the approaches of Koppel et al. (2011a), Akiva and Koppel (2013), Akiva and Koppel (2013)-SynonymSet and *Proposed-1*. The purity results of the approach in Akiva and Koppel (2012) are also used for comparison in Table 4.5.

As shown in Tables 4.4 and 4.5, the purity results of our approach using the documents created by merging two Bible books are quite promising and better than the other five state-of-the-art approaches.

It is worth to note that, unlike the proposed approach in this chapter, the approaches in Koppel et al. (2011a) and Akiva and Koppel (2013)-SynonymSet were developed for Bible books only, and not usable for other documents.

TABLE 4.5: Purity results of documents created by merging two bibles of different literatures. Approaches in comparison are noted as: [1] Koppel et al. (2011a), [2] Akiva and Koppel (2012), [3] Akiva and Koppel (2013), [4] Akiva and Koppel (2013)-SynonymSet, [5] *Proposed-1* and our approach.

Doc.	**[1]**	**[2]**	**[3]**	**[4]**	**[5]**	**Ours**
Job-Prov	84.5%	84.9%	93.9%	82.0%	95.2%	98.4%
Jer-Eze	82.0%	87.6%	96.6%	95.9%	97.0%	97.1%
Isa-Jer	71.8%	63.4%	66.7%	82.7%	71.0%	73.6%
Isa-Eze	78.9%	76.0%	80.0%	88.0%	82.7%	84.9%
Overall	***79.3%***	***78.0%***	***84.3%***	***87.2%***	***86.5%***	***88.5%***

4.5.2.3 Results on Scientific Document

In order to demonstrate that the proposed approach can also work with authentic documents, we have applied the approach on a very early draft of a scientific paper written by two Ph.D students (i.e., Author 1 and Author 2) in our research group. Authors 1 and 2 have contributed 41.9% and 58.1% of the document, respectively. Table 4.6 shows the numbers of sentences that are correctly classified according to the authorship using the proposed approach. Furthermore, Table 4.6 also presents the predicted contribution of each of the two authors (Author 1 and Author 2) using the approach. As shown in Table 4.6, the proposed approach achieves an overall purity of 93.0% for the scientific document. For comparisons, the purity result of the same document using our previously proposed approach (i.e., $Proposed-1$) is 92.0%.

From Table 4.6, it can be seen that the purity results and the predicted contribution of each author using the proposed approach on an authentic document are very promising. This shows that our approach can also be applied to genuine documents and define the authors' contributions in a multi-author document.

4.6 Summary

In this chapter, an unsupervised, hierarchical learning framework for segmenting a multi-author document into authorial components has been presented. We started from estimating the writing styles reflected by consecutive number of sentences and generated the

TABLE 4.6: The purity results and predicted contributions of two authors of a scientific paper obtained using the proposed approach.

Author	**No. of Sentences**	**Correctly Classified Sentences**	**Purity Result**	**Predicted Contribution**
A	131	123	93.9%	43.8%
B	182	168	92.3%	56.2%
Overall Purity		**93.0%**		

initial class information of data. Then, we took advantage of the difference in the posterior probabilities of the Naive-Bayesian model and created meaningful training datasets with a high precision for accurate supervised learning. The proposed approach has been evaluated in documents when two or more authors have cooperated in writing them. We have examined our approach on 28 multi-author documents created by using three very well-known benchmark datasets, of which each has its own focus and characteristics. The results of the proposed approach surpass those using the state-of-the-art approaches shown in Koppel et al. (2011a), Akiva and Koppel (2012), Akiva and Koppel (2013), Giannella (2015) and *Proposed-1* in terms of purity result. Single-topic documents have also been tested to verify that our approach tends to segment the document according to the authorship, rather than topics. Furthermore, a scientific paper has been used to show the application of the proposed approach to the authorship segmentation on an authentic document.

For brevity, in the following chapters of this thesis, we denote the proposed approach presented in this chapter by *Proposed-2*.

The proposed approach presented in Chapter 3 (i.e., *Proposed-1*), as well as the proposed approach presented in this chapter (i.e., *Proposed-2*) do not employ the contextual information hidden among sentences of the document for segmenting the sentences into authorial components. They consider only the information embedded in a sentence to find its authorial information without exploiting the information carried in other sentences. In next chapter, a new approach of authorship-based multi-author document segmentation is proposed. The approach will utilize the useful sequential correlation among the sentences in order to determine the authorial components.

Chapter 5

Unsupervised Multi-Author Document Decomposition Based on Hidden Markov Model

Chapters 3 and 4 have presented new approaches for authorship-based multi-author document decomposition where a Bayes' theorem with strong (naive) independence assumptions has been used. The experimental results have shown that the proposed approaches achieve interesting performance in decomposing a multi-author document based on authorship.

This chapter proposes a new unsupervised approach for segmenting a multi-author document into authorial components. The key novelty is that we utilise the sequential patterns hidden among document elements when determining their authorships. For this purpose, we adopt Hidden Markov Model (HMM) and construct a sequential probabilistic model to capture the dependencies of sequential sentences and their authorships. An unsupervised learning method is developed to initialise the HMM parameters. Experimental results on benchmark datasets have demonstrated the significant benefit of our idea and our approach has outperformed the state-of-the-arts on all tests. As an example of its applications, the proposed approach is applied for attributing authorship of a document and has also shown promising results.

5.1 Introduction

Authorship analysis is a process of inspecting documents in order to extract authorial information about these documents. It is considered as a general concept that embraces several types of authorship subjects, including *authorship verification*, *plagiarism detection* and *author attribution*. Each subject differs from others in the type of authorship information to be extracted from a document. Authorship verification (Brocardo et al., 2013; Potha and Stamatatos, 2014) decides whether a given document is written by a specific author. Plagiarism detection (Stein et al., 2011; Kestemont et al., 2011) seeks to expose the similarity between two texts. However, it is unable to determine whether they are written by the same author. In author attribution (Juola, 2006; Savoy, 2016), a real author of an anonymous document is predicted using labeled documents of a set of candidate authors.

Another significant subject in authorship analysis, which has received comparatively less attention from research community, is *authorship-based multi-author document decomposition*. This subject is to group the sentences of a multi-author document to different classes, of which each contains the sentences written by only one author. Many applications can take advantage of such a subject, especially those in forensic investigation, which aim to determine the authorship of sentences in a multi-author document. Furthermore, this kind of subject is beneficial for detecting plagiarism in a document and defining contributions of authors in a multi-author document for commercial purpose. Authorship-based multi-author document decomposition can also be applied to identify which source (regarded as an 'author' in this work) a part of a document is copied from when the document is formed by taking contents from various sources.

Despite of the benefits of authorship-based multi-author document decomposition, there has been little research reported on this subject. Koppel et al. (2011a) are the first researchers who implemented an unsupervised approach for decomposing a document into authorial components. However, their approach is restricted to Hebrew documents only. The authors of Akiva and Koppel (2013) addressed the drawbacks of the above approach by proposing a generic unsupervised approach for authorship-based multi-author document decomposition. Their approach utilised distance measurements to increase the

precision and accuracy of clustering and classification phases, respectively. They implemented their approach in two different ways. The first one, usable for all document types, employed a feature set containing the most frequent 500 words appearing in the corresponding document to represent the document. The second one, usable for particular documents only (e.g., Bible books), employed a feature set containing synonyms to represent the corresponding document. The accuracy of their approach is highly dependent on the number of authors. When the number of authors increases, the accuracy of the approach is significantly dropped. Giannella (2015) presented an improved approach for authorship-based multi-author document decomposition when the number of authors of the document is known or unknown. In his approach, a Bayesian segmentation algorithm is applied, which is followed by a segment clustering algorithm. However, the author tested his approach by using only documents with a few transitions among authors. Furthermore, the accuracy of the approach is very sensitive to the setting of its parameters.

In Chapters 3 and 4, we have proposed two unsupervised multi-author document decomposition approaches, i.e., *Proposed-1* and *Proposed-2* respectively, by exploiting the differences in the posterior probabilities of a Naive-Bayesian model in order to increase the purity result, and to be less dependent on the number of authors compared with the approach in Akiva and Koppel (2013). Our works have been tested on documents with up to 400 transitions among authors and the purity results of the approaches are not sensitive to the setting of parameters, in contrast with the approach in Giannella (2015). However, the performance of these approaches greatly depend on a threshold, of which the optimal value for an individual document is not easy to find.

Some other works have focused on segmenting a document into components according to their topics. For applications where the topics of documents are unavailable, these topic-based solutions will fail. In this chapter, the document decomposition approach is independent of documents' topics.

All of the existing works have assumed that the observations (i.e., sentences) are independent and identically distributed (iid). No consideration has been given to the

contextual information between the observations. However, in some cases, the iid assumption is deemed as a poor one (Rogovschi et al., 2010). In this chapter, we will relax this assumption (i.e., iid) and consider sentences of a document as a sequence of observations. We make use of the contextual information hidden between sentences in order to identify the authorship of each sentence in a document. In other words, the authorships of the "previous" and "subsequent" sentences have relationships with the authorship of the current sentence. Therefore, in this chapter, a well-known sequential model, Hidden Markov Model (HMM), is used for modelling the sequential patterns of the document in order to describe the authorship relationships. According to literature review, the HMM is applied to process a sequential data in many various areas, such as speech recognition (Abdel-Hamid et al., 2012; Debyeche et al., 2014), text recognition (Espana-Boquera et al., 2011; Roy et al., 2013) and biological analysis (Baldi and Brunak, 2001; Wheeler et al., 2013). However, no HMM-based approaches have been found to address the problem of multi-author document decomposition.

The contributions of this chapter are summarised as follows.

1. We capture the dependencies between consecutive elements in a document to identify different authorial components and construct an HMM for classification. It is for the first time the sequential patterns hidden among document elements is considered for such a problem.

2. To build and learn the HMM model, an unsupervised learning method is first proposed to estimate its initial parameters, and it does not require any information of authors or document's context other than how many authors have contributed to write the document.

3. Different from the *Proposed-1* and *Proposed-2* approaches presented in Chapters 3 and 4 respectively, the proposed unsupervised approach no longer relies on any predetermined threshold for authorship-based multi-author document decomposition.

4. Comprehensive experiments are conducted to demonstrate the superior performance of our ideas on both widely-used artificial benchmark datasets and an

authentic scientific document. As an example of its applications, the proposed approach is also applied for attributing authorship on a popular dataset. The proposed approach can not only correctly determine the author of a disputed document but also provide a way for measuring the confidence level of the authorship decision for the first time.

The rest of this chapter is organised as follows. Section 5.2 presents the framework of the proposed approach. Section 5.3 presents the initialising process of HMM parameters. Section 5.4 discusses the learning process of HMM parameters. Section 5.5 describe the Viterbi decoding process. Experiments are conducted in Section 5.6. Finally, Section 5.7 presents the summary of the chapter.

5.2 Framework of the Proposed Approach

The authorship-based multi-author document decomposition can be formulated as follows. Given a multi-author document C, written by N co-authors, it is assumed that each sentence in the document is written by one of the N co-authors. Furthermore, each co-author has written long successive sequences of sentences in the document. The number of authors N is known beforehand, while typically no information about the document contexts and co-authors is available. Our objective is to define the sentences of the document that are written by each co-author.

Our approach consists of three steps shown as follows.

1. Estimate the initial values of the HMM parameters $\{\boldsymbol{\pi}, \boldsymbol{B}, \boldsymbol{A}\}$ with a novel unsupervised learning method.
2. Learn the values of the HMM parameters using the $Baum - Welch$ algorithm (Baum, 1972; Bilmes et al., 1998).
3. Apply the $Viterbi$ algorithm (Forney Jr, 1973) to find the most likely authorship of each sentence.

In the following three sections, we discuss these three main steps in more detail.

5.3 Initializing Parameters of HMM

In our approach, we assume that we do not know anything about the document C and the authors, except for the number of co-authors of the document (i.e., N). This approach applies an HMM in order to classify each sentence in document C into a class corresponding to its co-author. The step (see Section 5.4) for learning of HMM parameters $\{\boldsymbol{\pi}, \boldsymbol{B}, \boldsymbol{A}\}$ is heavily dependent on the initial values of these parameters (Wu, 1983; Xu and Jordan, 1996; Huda et al., 2006). Therefore, a good initial estimation of the HMM parameters can help achieve a higher classification result.

We take advantage of the sequential information of data and propose an unsupervised approach to estimate the initial values of the HMM parameters. The detailed steps of this approach are shown as follows.

1. The document C is divided into *segments*. Each segment has 30 successive sentences, where the i^{th} segment comprises the i^{th} 30 successive sentences of the document. This will produce s segments, where $s = \text{Ceiling}(|C|/30)$ with $|C|$ represents the total number of sentences in the document. The number of sentences in each segment (i.e., 30) is chosen in such a way that each segment is long enough for representing a particular author's writing style, and also the division of the document gives an adequate number of segments in order to be used later for estimating the initial values of HMM parameters.

2. We select the words appearing in the document for more than two times. This produces a set of D words. For each segment, create a D-dimensional vector where the i^{th} element in the vector is one (zero) if the i^{th} element in the selected word set does (not) appear in the segment. Therefore, s binary D-dimensional vectors are generated, and the set of these vectors is denoted by $X = \{x_1, \cdots, x_s\}$.

3. A multivariate Gaussian Mixture Models (GMMs) (McLachlan and Peel, 2004) is used to cluster the D-dimensional vectors X into N components denoted by $\{r_1, r_2, \cdots, r_N\}$. Note that the number of components is equal to the number of co-authors of the document. Based on the GMMs, each vector, x_i, gets a

label representing the Gaussian component that this vector x_i is assigned to, for $i = 1, 2, \cdots, s$. The GMMs clustering technique is discussed in Section 2.4.

4. Again, we represent each segment as a binary vector using a new feature set containing all words appearing in the document for at least once. Assuming the number of elements in the new feature set is D', s binary D'-dimensional vectors are generated, and the set of these vectors is denoted by $X' = \{x'_1, \cdots, x'_s\}$. Each vector x'_i will have the same label of vector x_i, for $i = 1, 2, \cdots, s$.

5. We construct a Hidden Markov model with a sequence of observations O' and its corresponding sequence of hidden states Q'. In this model, O' represents the resulted segment vectors X' of the previous step. Formally, observation o'_i, is the i^{th} binary D'-dimensional vector x'_i, that represents the i^{th} segment of document C. In contrast, Q' represents the corresponding authors of the observation sequence O'. Each q'_i symbolises the most likely author of observation o'_i. According to Steps 3 and 4 of this section, each x'_i representing o'_i takes one label from a set of N elements, and the label represents its state, for $i = 1, 2, \cdots, s$.

   By assigning the most likely states to all hidden states (i.e., $q'_i, i = 1, 2, \cdots, s$), the state transition probabilities $\boldsymbol{A}$ are estimated.

   As long as there is only one sequence of states in our model, the initial probability of each state is defined as the fraction of times that the state appears in the sequence Q', so

$$\pi_n = \frac{Count(q' = r_n)}{Count(q')}, \quad for \quad n = 1, 2, \cdots, N. \tag{5.1}$$

6. Given the sequence X', and the set of all possible values of labels, the conditional probability of feature f^k in X' given a label r_n, $p(f^k|r_n)$, is computed, for $k = 1, 2, \cdots, D'$ and $n = 1, 2, \cdots, N$.

7. The document C is partitioned into sentences. Let $z = |C|$ represent the number of sentences in the document. We represent each sentence as a binary feature vector using the same feature set used in Step 4. Therefore, z binary D'-dimensional

vectors, denoted by $O = \{o_1, \cdots, o_z\}$, are generated. By using the conditional probabilities resulted in Step 6, the initial values of $\boldsymbol{B}$ are computed as

$$p(o_i|r_n) = \prod_{k=1}^{D'} o_i^{f^k} p(f^k|r_n), \tag{5.2}$$

where $o_i^{f^k}$ represents the value of feature f^k in sentence vector o_i, for $i = 1, 2, \cdots, z$ and $n = 1, 2, \cdots, N$.

In this approach, we use *add-one smoothing* (Martin and Jurafsky, 2000) for avoiding zero probabilities of $\boldsymbol{A}$ and $\boldsymbol{B}$. Furthermore, we take the logarithm function of the probability in order to simplify its calculations.

The initial values of the $\boldsymbol{A}$, $\boldsymbol{B}$ and $\boldsymbol{\pi}$ are now available. In next section, the learning process of these parameter values is performed.

5.4 Learning HMM

After estimating the initial values for the parameters of HMM (i.e., $\boldsymbol{\pi}$, $\boldsymbol{B}$ and $\boldsymbol{A}$), we now find the parameter values that maximise the likelihood of the observed data sequence (i.e., sentence sequence). The learning process of the HMM parameter values is performed as follows.

1. Construct a Hidden Markov model with a sequence of observations, O, and a corresponding sequence of hidden states, Q. In this model, O represents the resulted sentence vectors (Step 7 in the previous section). Formally, the observation o_i, is the i^{th} binary D'-dimensional vector and it represents the i^{th} sentence of document C. In contrast, Q represents the corresponding authors of observation sequence O. Each q_i symbolizes the most likelihood author of observation o_i, for $i = 1, 2, \cdots, z$

2. The Baum-Welch algorithm is applied to learn the HMM parameter values. The algorithm, also known as the $forward - backward$ algorithm (Rabiner, 1989), has two steps, i.e., *E-step* and *M-step*. The *E-step* finds the expected author sequence

(Q) of the observation sequence (O), and the *M-step* updates the HMM parameter values according to the state assignments. The learning procedure starts with the initial values of HMM parameters, and then the cycle of these two steps continues until a convergence is achieved in $\boldsymbol{\pi}$, $\boldsymbol{B}$ and $\boldsymbol{A}$. More details about the Baum-Welch algorithm can be found in Subsection 2.3.3.

The learned HMM parameter values will be used in the next section in order to find the best sequence of authors for the given sentences.

5.5 Viterbi Decoding

For a Hidden Markov model, there are more than one sequence of states in generating the observation sequence. The Viterbi decoding algorithm (Forney Jr, 1973) is used to determine the best sequence (i.e., best path) of states for generating observation sequence. Therefore, by using the Hidden Markov model that is constructed in previous section and the learned HMM parameter values, the Viterbi decoding algorithm is applied to find the best sequence of authors for the given sentences. More details about the Viterbi algorithm can be found in Subsection 2.3.4.

5.6 Experiments

In this section, we demonstrate the performance of our proposed approach by conducting experiments on benchmark datasets as well as one authentic document. Furthermore, an application on authorship attribution is presented using another popular dataset.

5.6.1 Datasets

Three benchmark corpora widely used for authorship analysis (i.e., Bible books, Becker-Posner blogs and *New York Times* articles) and an authentic document are used to evaluate our approach. The descriptions of these four corpora are found in Chapter 3

(i.e., Section 3.7). Furthermore, the six single-topic documents of Becker-Posner blogs that are described in Chapter 4 (i.e., Section 4.5) are also examined in this chapter.

5.6.2 Experimental Results on Document Decomposition

The performance of the proposed approach is evaluated through a set of comparisons with four state-of-the-art approaches on the four aforementioned datasets.

The experiments on the first three datasets (i.e., Bible books, Becker-Posner blogs and *New York Times* articles), excluding the six single-topic documents, are conducted using a set of artificially merged multi-author documents. These documents are created by using the same method that has been used in Chapter 3 (i.e., Section 3.2).

5.6.2.1 Results on the Biblical Books Dataset

We utilise the Bible books of five authors and create artificial documents by merging books of any two possible authors. This produces 10 multi-author documents of which six have different types of literatures and four have the same type of literature. Tables 5.1 and 5.2 show the comparisons of purity results of the documents composed by merging two biblical books of different literatures (Table 5.1) and the documents composed by merging two biblical books of the same literatures (Table 5.2), respectively, using our approach and the approaches developed by Koppel et al. (2011a), Akiva and Koppel (2013)-500CommonWords, Akiva and Koppel (2013)-SynonymSet and *Proposed-1*. The purity results of the approach in Akiva and Koppel (2012) are used further for comparison in Table 5.2.

As shown in Tables 5.1 and 5.2, the results of our approach are very promising. The overall purities of documents of different literatures or the same literatures are better than the other five state-of-the-art approaches.

In our approach, we have proposed an unsupervised method to estimate the initial values of the HMM parameters (i.e., $\boldsymbol{\pi}$, $\boldsymbol{B}$ and $\boldsymbol{A}$) using segments. Actually, the initial values of the HMM parameters are sensitive factors to the convergence and accuracy of the learning process. Most of the previous works using HMM have estimated these values

TABLE 5.1: Purity results of merged documents of *different literature* bible books using the approaches of 1- Koppel et al. (2011a), 2- Akiva and Koppel (2013)-500CommonWords, 3- Akiva and Koppel (2013)-SynonymSet, 4- *Proposed-1* and 5- our approach.

Doc.	**1**	**2**	**3**	**4**	**5**
Eze-Job	85.8%	98.9%	95.0%	99.0%	99.4%
Eze-Prov	77.0%	99.0%	91.0%	98.0%	98.8%
Isa-Prov	71.0%	95.0%	85.0%	98.0%	98.7%
Isa-Job	83.0%	98.8%	89.0%	99.0%	99.4%
Jer-Job	87.2%	98.2%	93.0%	98.0%	98.5%
Jer-Prov	72.2%	97.0%	75.0%	99.0%	99.5%
Overall	***79.4%***	***97.8%***	***88.0%***	***98.5%***	***99.1%***

TABLE 5.2: Purity results of merged documents of the *same literature* bible books using the approaches of 1- Koppel et al. (2011a), 2- Akiva and Koppel (2012), 3- Akiva and Koppel (2013)-500CommonWords, 4- Akiva and Koppel (2013)-SynonymSet, 5- *Proposed-1* and 6- our approach.

Doc.	**1**	**2**	**3**	**4**	**5**	**6**
Job-Prov	85.0%	84.9%	94.0%	82.0%	95.0%	98.2%
Isa-Jer	72.0%	63.4%	66.9%	82.9%	71.0%	72.1%
Isa-Eze	79.0%	76.0%	80.0%	88.0%	83.0%	83.2%
Jer-Eze	82.0%	87.6%	97.0%	96.0%	97.0%	97.3%
Overall	***79.5%***	***78.0%***	***84.5%***	***87.2%***	***86.5%***	***87.7%***

by clustering the original data, i.e., they have clustered sentences rather than segments. Figure 5.1 compares the results of using segments with the results of using sentences for estimating the initial parameters of HMM in the proposed approach for the 10 merged Bible documents in terms of the purity results and number of iterations till convergence, respectively. From Figures 5.1, one can notice that the purity results obtained by using segments for estimating the initial HMM parameters are significantly higher than using sentences for all merged documents. Furthermore, the number of iterations required for convergence for each merged document using segments is significantly smaller than using sentences.

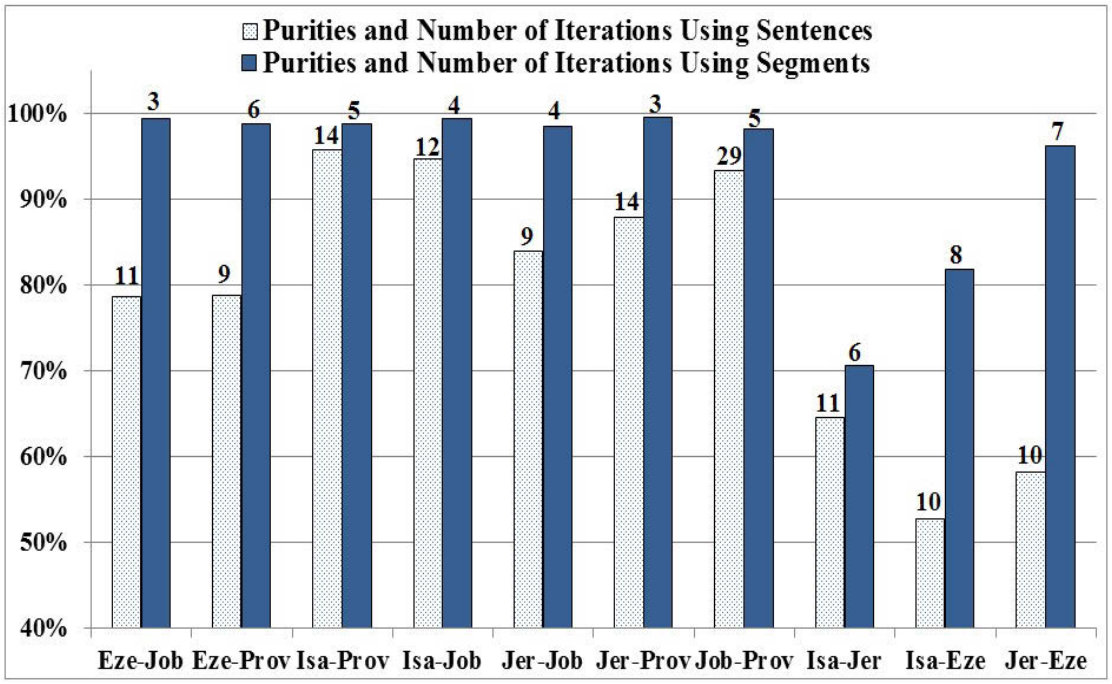

FIGURE 5.1: Comparisons between using segments and using sentences in the unsupervised method for estimating the initial values of the HMM of our approach in terms of purity (representd as the cylinders) and number of iterations required for convergence (represented as the numbers above cylinders) using the 10 merged Bible documents.

5.6.2.2 Results on Becker-Posner Blogs Dataset (Controlling for Topic)

In our experiments, we represent Becker-Posner blogs in two different terms. The first term is as in Akiva and Koppel (2013) approach, where the whole blogs are exploited to create one merged document. The resulted merged document contains 26,922 sentences and more than 240 switches between the two authors. We obtain a purity of 96.72% when testing our approach in the merged document. The obtained result of such type of document, which does not have topic indications to differentiate between authors, is delightful. The first set of cylinders labelled "Becker-Posner" in Figure 5.2 shows the comparisons of purity results of our approach and the approaches of Akiva and Koppel (2013) and *Proposed-1* when the whole blogs are used to create one merged document. As shown in Figure 5.2, our approach yields better purity result than the other two approaches.

The second term is as in the approach of Giannella (2015), where six merged single-topic documents are formed. Due to comparatively shorter lengths of these documents, the

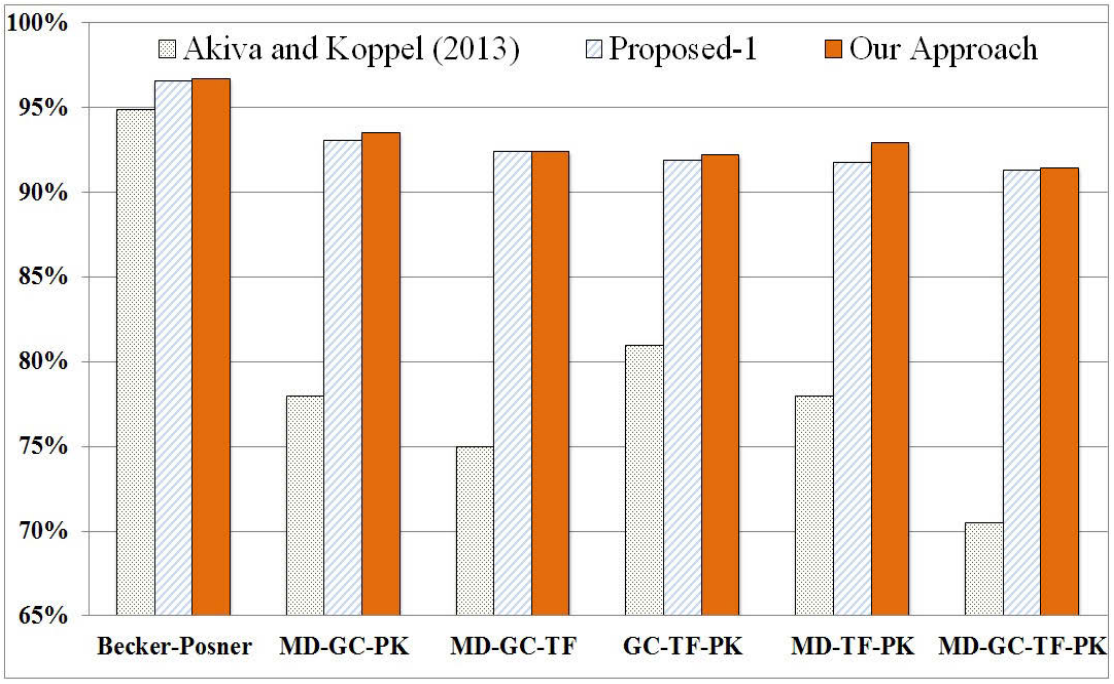

FIGURE 5.2: Purity comparisons between our approach and the approaches presented in Akiva and Koppel (2013) and *Proposed-1* in Becker-Posner documents, and documents created by three or four *New York Times* columnists (TF = Thomas Friedman, PK = Paul Krugman, MD = Maureeen Dowd, GC = Gail Collins).

number of resulted segments that are used for the unsupervised learning in Section 5.3 is clearly not sufficient. Therefore, instead of splitting each document into segments of 30 sentences length each, we split it into segments of 10 sentences length each. Figure 5.3 shows the purity results of the six documents using our approach and the approach presented in Giannella (2015). It is observed that our proposed approach has achieved higher purity result than Giannella (2015) in all of the six documents.

5.6.2.3 Results on *New York Times* Articles Dataset (N $\geq$ 2)

We perform our approach on *New York Times* articles. These articles are written by four columnists. For this corpus, the experiments can be classified into three groups. The first group is for those merged documents that are created by combining articles of any pair of the four authors. The six resulted documents have on average more than 250 switches between authors. The purity results of these documents are between 93.9% and 96.3%. It is notable that the results are very satisfactory for all documents. For comparisons, the purity results of the same documents using the *Proposed-1* approach

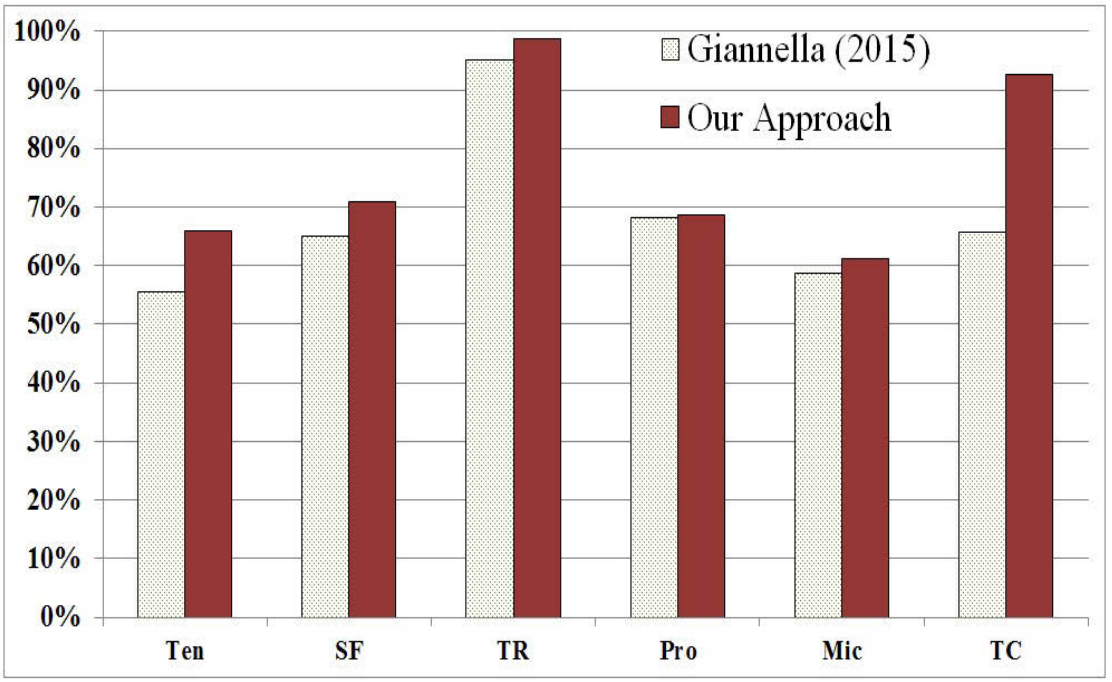

FIGURE 5.3: Purity comparisons between our approach and the approach presented in (Giannella, 2015) in the six single-topic documents of Becker-Posner blogs.

range from 93.3% to 96.1%. Furthermore, some of these documents have produced a purity lower than 89.0% using the approach of Akiva and Koppel (2013).

The second group is for those merged documents that are created by combining articles of any three of the four authors. The four resulted documents have on average more than 350 switches among the authors. The third group is for the document that are created by combining articles of all four columnists. The resulted merged document has 46,851 sentences and more than 510 switches among authors. Figure 5.2 shows the purity results of the five resulted documents regarding the experiments of the last two groups. Furthermore, it shows the comparisons of our approach and the approaches presented in Akiva and Koppel (2013) and *Proposed-1*. It is noteworthy that the purities of our approach are better than the other two approaches in all of the five documents.

5.6.2.4 Results on Scientific Document

In order to demonstrate that our proposed approach is applicable on genuine documents as well, we have applied the approach on first draft of a scientific paper written by two

TABLE 5.3: The purity results and predicted contributions of the two authors of the scientific paper using the proposed approach.

Author	**Purity Result**	**Predicted Contribution**
1	98.5%	47.6%
2	89.0%	52.4%
Purity	**93.0%**	

Ph.D. students (Author 1 and Author 2) in our research group. Each student was assigned a task to write some full sections of the paper. Author 1 has contributed 41.9% of the document and Author 2 contributed 58.1%. Table 5.3 shows the number of correctly assigned sentences of each author and the purity resulted using the proposed approach. Table 5.3 also displays the authors' contributions predicted using our approach. As shown in Table 5.3, the proposed approach has achieved an overall purity of 93.0% for the authentic document.

5.6.3 Experimental Results on Authorship Attribution

One of the applications that can take advantage of the proposed approach is the authorship attribution (i.e., determining a real author of an anonymous document given a set of labeled documents of candidate authors). The *Federalist Papers* dataset have been employed in order to examine the performance of our approach for this application. This dataset is considered as a benchmark in authorship attribution task and has been used in many studies related to this task (Juola, 2006; Savoy, 2013b, 2016). The *Federalist Papers* consist of 85 articles published anonymously between 1787 and 1788 by Alexander Hamilton, James Madison and John Jay to persuade the citizens of the State of New York to ratify the Constitution. Of the 85 articles, 51 of them were written by Hamilton, 14 were written by Madison and 5 were written by Jay. Furthermore, 3 more articles were written jointly by Hamilton and Madison. The other 12 articles (i.e., articles 49-58 and 62-63), the famous "anonymous articles", have been alleged to be written by Hamilton or Madison.

To predict a real author of the 12 anonymous articles, we use the first five undisputed articles of both authors, Hamilton and Madison. Note that we ignore the articles of Jay

because the anonymous articles are alleged to be written by Hamilton or Madison. The five articles of Hamilton (articles 1 and 6-9) are combined with the five articles of Madison (articles 10, 14 and 37-39) in a single merged document where all the articles of Hamilton are inserted into the first part of the merged document and all the articles of Madison are inserted into the second part of the merged document. The merged document has 10 undisputed articles covering eight different topics (i.e., each author has four different topics). Before applying the authorship attribution on the 12 anonymous articles, we have tested our approach on the resulted merged document and a purity of 95.2% is achieved in this document. Note that, the authorial components in this document are not thematically notable.

For authorship attribution of the 12 anonymous articles, we add one anonymous article each time on the middle of the merged document, i.e., between Hamilton articles part and Madison articles part. Then, we apply our approach on the resulted document, which has 11 articles, to determine to which part the sentences of the anonymous article are classified to be sentences of Hamilton or Madison. As the ground truth for our experiments, all of these 12 articles can be deemed to have been written by Madison because the results of all recent state-of-the-art studies testing on these articles on authorship attribution have classified the articles to Madison's. Consistent with the state-of-the-art approaches, these 12 anonymous articles are also correctly classified to be Madison's using the proposed approach. Actually, all sentences of articles 50,52-58 and 62-63 are classified as Madison's sentences, and 81% of the sentences of article 49 and 80% of article 51 are classified as Madison's sentences. Table 5.4 presents the number of sentences that are classified as Madison sentences and Hamilton sentences of each of the 12 anonymous articles As seen from Table 5.4, all the sentences of articles 50,52-58 and 62-63 are classified as Madison's sentences, . These percentages can be deemed as the confidence levels (i.e., 80% confidences for articles 49, 81% for 51, and 100% confidences for all other articles) in making our conclusion of the authorship contributions.

TABLE 5.4: The number of sentences that are classified with Madison sentences and Hamilton sentences of each of the 12 anonymous articles of *The Federalist Papers* using the proposed approach.

Article	**Number of Sentences**	**Classified Madison**	**Classified Hamilton**
# 49	57	46	11
# 50	41	41	0
# 51	61	49	12
# 52	61	61	0
# 53	68	68	0
# 54	64	64	0
# 55	61	61	0
# 56	47	47	0
# 57	78	78	0
# 58	61	61	0
# 62	75	75	0
# 63	85	85	0

5.7 Summary

A new unsupervised approach for decomposing a multi-author document into authorial components has been developed. Different from the state-of-the-art approaches, we have innovatively made use of the sequential information hidden among document elements. For this purpose, we have used HMM and constructed a sequential probabilistic model, which is used to find the best sequence of authors that represents the sentences of the document. An unsupervised learning method has also been developed to estimate the initial parameter values of HMM. Comparative experiments conducted on benchmark datasets have demonstrated the effectiveness of our ideas with superior performance achieved on both artificial and authentic documents. An application of the proposed approach on authorship attribution has also achieved perfect results of 100% purity results together with confidence measurement for the first time.

In the next chapter, the proposed approach presented in this chapter will be further extended and a two-stage HMM model for utilizing the sequential patterns among sentences more efficiently will be presented.

For brevity, in the following chapters of this thesis, we denote the proposed approach presented in this chapter by *Proposed-3*.

Chapter 6

SUDMAD: Sequential and Unsupervised Decomposition of a Multi-Author Document Based on a Hidden Markov Model

Decomposing a multi-author document into sentences based on authorship is of great interest due to the increasing demand for many different applications, such as plagiarism detection, forensic analysis, civil law and intelligence issues that involve disputed anonymous documents. Among existing studies for document decomposition some were limited by specific languages, according to topics or restricted to a document of two authors, and their accuracies have big rooms for improvement. In this chapter, we consider the contextual correlation hidden among sentences and propose an algorithm for Sequential and Unsupervised Decomposition of a Multi-Author Document (SUDMAD) written in any language disregarding to topics, through the construction of a Hidden Markov Model (HMM) reflecting authors' writing styles. Note that, the initial idea of this work has been presented in Chapter 5. A simple HMM was constructed to find a useful sequential correlation between consecutive sentences of the document, which has achieved very encouraging results. In this chapter, we further extend our work and propose a two-stage HMM model in order to utilise the sequential patterns among

sentences more comprehensively. To build and learn such a model, an unsupervised, statistical approach is first proposed to estimate the initial values of HMM parameters of a preliminary model, which does not require the availability of any information of authors or document's context other than how many authors have contributed to writing the document. To further boost the performance of this approach, a boosted HMM learning procedure is proposed next, where the initial classification results are used to create labelled training data to learn a more accurate HMM. Moreover, the contextual relationship among sentences is further utilised to refine the classification results. Our proposed approach is empirically evaluated on four benchmark datasets which are widely used for authorship analysis of documents. Comparisons with recent state-the-art approaches are also presented to demonstrate the significance of our new ideas and the superior performance of our approach.

6.1 Introduction

Authorship analysis is the process of analysing the authors of a disputed anonymous document, which uses a statistical study of linguistic called stylometry (Baayen et al., 2002) to identify the background of authors of the questioned text document. The task of authorship analysis is considered as a very old research topic. The first endeavor for identifying the writing style of a text document was in the 19th century with the study of Mendenhall (1887) on the Shakespeare's plays. Several studies in the 20th century have also focused on analysing a text document by exploiting measurements of some stylometric features in order to determine the author's writing style of the document (Zipf, 1932; Sheldon, 1991; Holmes and Forsyth, 1995; Holmes, 1998).

In recent years, authorship analysis has received increasing attention and been considered as an important problem in many fields including information retrieval and computational linguistics. This importance springs from the fact that the large amount of disputed information of documents on Internet needs to be analysed and investigated. Existing approaches on authorship analysis have focused on analysing texts with different formats, such as literature (McDowell and Melvin, 1983; Burrows et al., 2002; Juola, 2006), online messages (e.g., email, blogs) (Abbasi and Chen, 2005; Zheng et al., 2006;

Nirkhi et al., 2012) and program codes (Krsul and Spafford, 1995; Rosenblum et al., 2011; Burrows et al., 2014).

Many different scenarios have been considered for studying the authorship analysis. For example, the works of Koppel and Schler (2004) and Koppel and Winter (2014) focused on the authorship verification problem, also called similarity detection problem (El and Kassou, 2014). They aimed to determine whether two documents were written by the same author without the attention of the real author. In this case, there is no need to have a set of candidate authors. Another important scenario, which has been studied extensively in the last few years, is the authorship attribution (Koppel et al., 2009b; Stamatatos, 2009a; Luyckx and Daelemans, 2011; Koppel et al., 2013; Savoy, 2016). The idea is that, given text samples of a number of candidate authors, we are required to determine which of them is the real author of a given disputed text document.

In this chapter, we address another intriguing application scenario, which is also related to the authorship analysis, called "authorship-based multi-author document decomposition". The trajectory of this scenario is to decompose a document written by more than one author into components each written by only one author. Although this problem is very important because of applications in plagiarism detection (Stamatatos, 2011), forensic analysis (Orebaugh et al., 2014), civil law (i.e., disputed copyright issues) (Grant, 2007) and intelligence issues (Layton et al., 2010), studies on this area have been extremely limited so far. The work in Koppel et al. (2011a) has considered a new unsupervised approach for decomposing a multi-author document into authorial parts. They created artificially merged documents by using only one dataset containing 5 biblical books, which were written in Hebrew by 5 authors. However, this approach is limited to a specific type of documents only (i.e., Hebrew language documents), and it has been tested using only documents formed by two authors. Akiva and Koppel (2012) presented an unsupervised approach for identifying distinct authorial components of a multi-author document. Unlike the approach described in Koppel et al. (2011a), this approach has been tested on documents written by 2, 3 and 4 authors respectively, and also it is a language-independent approach. However, the overall accuracy of this approach is not high enough. One year later, this approach was further improved in Akiva and Koppel (2013) by taking advantages of distance-based methods. However, when

the number of authors increased to more than 2, the accuracy degraded significantly. For the same purpose, the approach was examined in Giannella (2015) and an improved approach called BayesAD was proposed, where the number of authors of the document can be either known or unknown. However, only documents with very few turns among authors were tested in the work, and its performance heavily relied on the parameter setting.

In Daks and Clark (2016), the authors have proposed an unsupervised approach for segmenting documents according to their authorships. However, they have assumed that each document has been written by only a single author.

Some researchers have investigated the problem of decomposing documents according to topics rather than authors (Beeferman et al., 1999; Allan, 2012; Jameel and Lam, 2013). This problem is quite different from that of this chapter. In fact, in at least two of our experiments, the topics among authors in a document are undifferentiable. Furthermore, topic-based decomposition approaches cannot handle single-topic documents, which will also be examined in this chapter.

Other researchers have focused on the task of text intrinsic plagiarism detection. The task, which has been directly addressed in PAN 2011 competition (Oberreuter et al., 2011; Kestemont et al., 2011; Rao et al., 2011), aims to determine whether a given suspicious document contains plagiarized text or not when no reference documents are provided. Furthermore, it detects plagiarized text in case that the document has a plagiarism. Most algorithms in intrinsic plagiarism detection attempt to detect plagiarized passages by analysing style changes within the document. Unlike the task of this chapter, in intrinsic plagiarism detection, usually most sentences of the document are written by one author (i.e., the main author) with limited percentage of the document written by other authors of which the number is not known. Whereas in the task that our work targets, each author has written long successive sentences in a document.

Some other researchers, such as Brooke et al. (2012), have presented a model for automatically segmenting a stylistically inconsistent text, i.e., identifying the points in a "multi-personal" poem The Waste Land (1922) by T. S. Eliot, where the style changes.

The work in Brooke et al. (2013) has also considered an unsupervised approach to distinguish voices in the same poem.

Typically, classical learning models are considered for constructing a classifier that can accurately predict the labels of new data given some training data. The main assumption made with regard to these models is that the data are independently and identically distributed (iid) from an unknown probability distribution. For example, the works presented in Chapters 3 (i.e., *Proposed-1*) and 4 (i.e., *Proposed-2*) have presented approaches for authorship-based multi-author document decomposition task where it is assumed that sentences in a document are iid. In this chapter, instead of assuming that the data are iid, we propose a novel idea to make use of the sequence of the data, i.e., the contextual relationship between the sentences. These sequences provide valuable sequential correlations. Sequential patterns are of great practical importance for many computational linguistic applications (Bishop, 2006), where they have been employed to enhance the prediction accuracy of classifiers. For example, in handwriting English text recognition, if the classifier estimates that one letter is Q, then there is a very high chance that the next letter will be U (Dietterich, 2002). Other examples are speech recognition (Abdel-Hamid et al., 2012), gene data analysis (Krogh, 1997) and stock market prediction (Gupta and Dhingra, 2012).

In our work of this chapter, we propose to segment a multi-author document into components according to authorship. We consider the contextual information hidden among series of sentences and propose to use the Hidden Markov Model (HMM) to explore the sequential patterns in the document. The initial idea of this work has been addressed in Chapter 5 where a simple HMM was used to decompose sentences into authorial components. Apart from more details and experiments that are included to disclose the benefit of this work, this chapter distinguishes from our previous related work presented in Chapter 5 significantly in the following three new contributions:

- We propose to utilise the useful sequential correlations among the consecutive sentences in order to determine the authorial components and construct a two-stage Hidden Markov Model, called "SequentialUD" - Sequential Unsupervised Decomposition, to model the relationships between authorships and sentences.

- To further boost the performance of this approach, a boosted HMM learning procedure is proposed. The initial classification results obtained using the statistically learned and preliminary HMM are used to create a labelled training dataset to learn a more accurate HMM.
- Moreover, the contextual relationships among sentences are further utilized to refine the classification results and a refined version of the SequentialUD is proposed.

In summary, the new approach proposed in this chapter further exhibits the benefits of exploring the sequential patterns of sentences for analysing document's authorships. This approach is completely unsupervised and does not require the availability of any information of authors or document's context other than the number of authors of the document. It is effective even when the topics in the document are not distinguishable among authors. When the number of authors increases, the performance of this approach is still very satisfactory. To the best of our knowledge, there have been no similar ideas reported in the literature.

The following section (i.e., Section 6.2) presents the framework of our proposed SequentialUD approach. The detailed procedure of estimating the initial parameters and learning the preliminary HMM using our proposed statistical approach are given in Section 6.3. The preliminary HMM is then used for the initial sentence decoding. In Section 6.4, the predicted labels are then used to create a labeled dataset from the unlabelled input, which is used to learn the final, boosted HMM. Eventually, sentence classification results are produced. A refinement procedure based on a modified probability indication procedure is proposed to further improve the purity, detailed in Section 6.5. Then, the experiments are presented in Section 6.6, followed by a summary of the chapter in Section 6.7.

6.2 Framework of the Proposed SequentialUD Approach

The problem of authorship-based multi-author document decomposition can be more formally presented as follows. Suppose that there are N (a known number greater than 1) authors who have participated in creating a document C, each author has written long

successive sentences in the document and each sentence is written by only one author. The goal is to decompose the sentences in the document into components according to their authorship, so that all sentences in a component are written by only one author.

In our work, we propose a new approach to address this problem by making use of the sequential correlations among the sentences and develop an unsupervised, sequential approach for document decomposition, called SequentialUD. The Hidden Markov Model (HMM) is constructed to classify each sentence to a corresponding author. To learn the HMM with no labeled data available, we develop an unsupervised, statistical approach to estimate the initial parameters needed for learning the model and generate label data. The procedure for learning a boosted HMM is then proposed to use the labelled data to learn the final HMM for more accurate prediction. Moreover, we modify our works in Chapters 3 and 4 and propose a modified version of the probability indication procedure (ModPIP) to further improve the purity results by considering the sequential patterns.

The framework of the proposed approach is shown in Figure 6.1. The modules enclosed by dashed lines represent the two stages of the proposed SequentialUD approach, i.e., Estimating the Preliminary HMM, and Learning the Boosted HMM. Optionally, the classification results can be refined to further improve its purity by performing ModPIP, resulting in a refined version of the SequentialUD approach.

As seen in Figure 6.1, our proposed SequentialUD approach has two main stages. In the first stage, given unlabelled input data, we first propose a statistical approach to estimate the initial parameters of a preliminary HMM, which enables the Baum-Welch Algorithm to learn the preliminary HMM. Once the preliminary HMM is learned, it is used to estimate the best sequence of authors for sentences in document C using the Viterbi Algorithm. With these initial prediction results, the approach then proceeds into Stage 2, where the problem now becomes a supervised learning problem to learn a boosted HMM. The predicted labels resulted from the first stage are now used to create a new, labelled training dataset, which is then used to learn a more accurate HMM. In the end, the Viterbi Algorithm is used again to find a more accurate sequence of authors for the sequence of all sentences of document C. As an optional step, the classification results can be further refined by taking use of the contextual information.

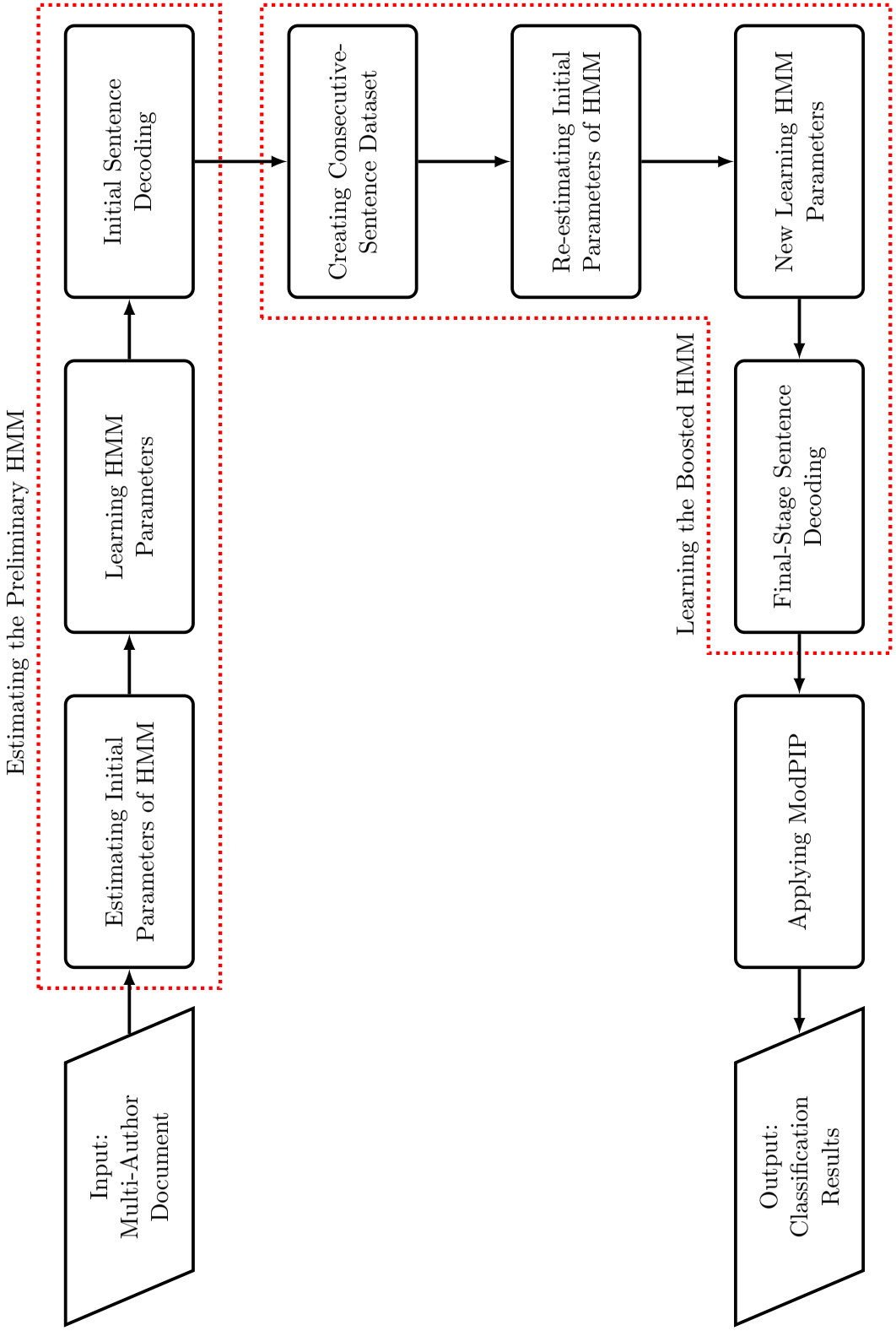

FIGURE 6.1: The framework of the proposed SequentialUD and its refined version.

6.3 Estimating a Preliminary HMM from Unlabelled Input Data

To make use of the contextual information for document decomposition, we utilise the Hidden Markov Model (HMM), a widely-used effective technique for sequential learning models, and take benefit from the powerful HMM tools to improve the classification purity result. In this section, we first briefly introduce the HMM. Then, we focus on how we formulate our document decomposition problem into the HMM and address the parameter initialization problem with no labelled data.

6.3.1 Hidden Markov Model

In HMM, the data are sequences of measurements and labels. These sequential data contain a wealth of precious information, that adjacent measurements and labels are expected to be related to each other, and can help to better grasp the underlying principles of many real-life problems. For our targeted document decomposition problem, where each author is assumed to have written long successive sentences in the document, intuitively, observing the authorship of one sentence is of great help for predicting the authorship of the next sentence.

The HMM is a statistical probabilistic model for sequential data consisting of a sequence of observable data and a hidden variable, which is not directly observable, for each observed data. The observable data are called "observations" and the hidden variables are called "hidden states". The hidden states in HMM form a Markov chain and the probability distribution of the observation depends on the underlying state.

Let us denote the T observations as $O = \{o_1, o_2, \ldots, o_T\}$ and the hidden states as $Q = \{q_1, q_2, \ldots, q_T\}$, where q_t is the hidden state of the t^{th} observation o_t. Each observation, which is assumed to be a discrete symbol, has one of the possible values from the set of observations $W = \{w_1, w_2, \ldots, w_M\}$ and each hidden state has one of the values from the set of states $S = \{s_1, s_2, \ldots, s_N\}$. Here, M and N represent the number of distinct observations and the number of distinct states in the model, respectively.

An HMM can be defined by three probabilities, $\boldsymbol{\theta} = \{\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{\pi}\}$, where $\boldsymbol{A}$ are the transition probabilities of all possible states, $\boldsymbol{B}$ are the emission probabilities of all observations given their states and $\boldsymbol{\pi}$ are the initial states probabilities. More details about HMM can be found in Section 2.3).

6.3.2 Estimating Initial Parameters of HMM

We consider HMM for document decomposition problem, where each observation represents one sentence and the hidden states represent the authors of the document. The goal is to decompose the document based on the writing style which is determined by the hidden state, i.e., authorship. The size of the observation and hidden state sequence, denoted by T, is the number of sentences in the document. In our model, due to that the number of distinct observations is not clearly observable, and the chance of having more than one sentence with the same syntactic structure is very low, we consider the number of unique observations (i.e., M) is also equal to the number of sentences in the document (i.e., T). Specifically, $T = M = |C|$ where $|C|$ is the number of sentences in document C. The number of unique states is equal to the number of authors of the document, which is denoted by N. The purpose of this model is to find the most probable sequence of authors that could have generated a given series of sentences in a document.

As illustrated in the previous subsection, an HMM can be specified by three parameters, $\boldsymbol{\theta} = \{\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{\pi}\}$. We learn this model by maximizing the likelihood function of HMM in order to find a best estimation of $\boldsymbol{\theta}$ so that the probability of the observations maximizes, as in

$$\theta = \arg\max_{\theta} \left(p(O|\theta)\right) \tag{6.1}$$

Normally, the learning process starts with some initial values of $\boldsymbol{\theta}$. For unsupervised learning problems (like the one we are dealing with), the initial values of $\boldsymbol{\theta}$ are not directly observed and therefore need to be manually set. The selection of $\boldsymbol{\theta}$ has a significant impact on the overall efficiency of the model as it directly affects the convergence rate

of the learning process, as well as whether the learning process can converge on global maximum (Zhang et al., 2007; Hoang and Hu, 2004).

In our work of this chapter, we propose a statistical approach and make use of the contextual information of sequential data to initiate the HMM parameter set $\boldsymbol{\theta}$. Next, we give the details of initializing these parameters in the order of transition matrix $\boldsymbol{A}$, prior $\boldsymbol{\pi}$, and emission probability $\boldsymbol{B}$. These are detailed as follows.

6.3.2.1 Estimating Transition Matrix $\boldsymbol{A}$

1. We first create a sequence of *segments*, where each segment is a series of v successive sentences from the document and does not overlap with any other segments. Intuitively, the segment length v relates to the length of the document, as well as the mean number of successive sentences in the document written by the same author. In the section of Experiments, detailed analysis is provided to find the most appropriate value of v for a given document. We then collect the statistic of the segments. Note that, working on segments instead of sentences allows us to capture the sequential patterns of sentences. Formally, let us denote the series of segments as $SEG = \{Seg_1, Seg_2, \ldots, Seg_e\}$. For a document of size $|C|$, this produces e segments, where $e = Ceiling(|C| \,/\, v)$. Notice that, each segment may be either a pure segment, where its sentences are produced by a single author, or a mixed segment, where its sentences are produced by more than one author.

2. For each segment, we then extract a feature vector based on the concept of "Bag of Words". To do this, first a word list is created for the document, where distinct words (i.e., the words occurred three or more times in the document) are added into a word list, denoted by $BagOfWords1 = \{word_1, word_2, \ldots, word_{D_1}\}$, where D_1 is the length of the list (i.e., the total number of the words in the list). In this thesis, a word is defined as a consecutive sequence of letters and digits. Then, each segment is represented as a D_1-dimension binary vector using the word list $BagOfWords1$, where each dimension takes a value of 1 or 0, with 1 indicating the corresponding word in the list appears in the segment and 0 indicating not. Thus, the segments SEG can be represented as a sequence of e D_1-dimension binary

feature vectors, denoted by $X = \{x_i, i = 1, 2, \ldots, e\}$. More details can be found in the Experiments section.

3. With the binary feature vectors X, we then cluster them into different groups, each representing a unique writing style. The Gaussian Mixture Model (GMM) (McLachlan and Basford, 1988) is adopted for clustering after comparing with classical clustering methods such as K-means (More details about GMM can be found in Section 2.4). Since there are N authors who have contributed writing the document C, the GMMs have N Gaussian components, each representing a different author's writing style. Each vector x_i, $i \in 1, 2, \ldots, e$, is clustered into one of the N Gaussian components.

4. Based on the Gaussian component that a vector x_i is assigned to during the above clustering process, each vector x_i is given a label. Apparently, the label of vector x_i, denoted by $h(x_i)$, takes one label from a set of N elements, i.e., $h(x_i) = n$, where $n \in \{1, 2, \ldots, N\}$.

   Note. The approaches of Akiva and Koppel (2012), Akiva and Koppel (2013), *Proposed-1* and *Proposed-2* also start from segmenting the original document into segments and then represent them as feature vectors in order to cluster them. However, the purpose of these steps in their approaches is different from the purpose in the proposed approach.

5. Then, with the labels $h(x_i)$, $i = 1, 2, \ldots, e$ of all the segments, the transition probability of moving from state n_1 to state n_2, denoted by $A_{n_1 n_2}$, can be computed using Eq. 6.2 as

$$A_{n_1 n_2} = \frac{Count\ (h(x_i) = n_2, h(x_{i-1}) = n_1) + 1}{Count\ (h(x_{i-1}) = n_1) + N}, \quad i = 2, \ldots, e, \tag{6.2}$$

   where $n_1, n_2 \in \{1, 2, \ldots, N\}$.

   Finding the transition probabilities of all possible state values (i.e., $n_1 = \{1, 2, \ldots, N\}$, $n_2 = \{1, 2, \ldots, N\}$) will produce the $N \times N$ transition matrix $\boldsymbol{A}$. Here, we employ the "add-1" smoothing technique (Manning and Schütze, 1999) in order to prevent zero values of transition probabilities.

6.3.2.2 the Prior $\boldsymbol{\pi}$

We then move on to estimate the initial probability $\pi(n)$, i.e., the prior probability of each author. With each segment $(x_i, i \in \{1, 2, \ldots, e\}$, where e is the number of feature vectors) being labelled as $h(x_i)$, the initial probability of each state, denoted by $\pi(n)$, can be simply measured as a fraction of the occurrences of each state $h(x)$ as: $\pi(n) = Count\ (h(x) = n)\ /\ e$, where $n \in \{1, 2, \ldots, N\}$.

Finding the initial probabilities of all possible state values (i.e., $n = \{1, 2, \ldots, N\}$) will produce a $1 \times N$ vector, which is denoted by $\boldsymbol{\pi}$.

6.3.2.3 Estimating the Emission Probabilities $\boldsymbol{B}$

The emission probabilities $\boldsymbol{B}$ address the relation between observations and states, i.e., given the authorship ("state"), the probability of observing each sentence ("observation").

1. The sequence of segments SEG, each consisting v successive sentences, is employed in order to find the initial value of $\boldsymbol{B}$. In order to do that, a new feature list, denoted as $BagOfWords2 = \{word_1, word_2, \ldots, word_{D_2}\}$, where D_2 is the length of the list, is created. The words that have occurred at least two times in the document are considered for this feature. The list of words are used for representing the sequence of segments (SEG) as a sequence of binary feature vectors, $X' = \{x'_i, i = 1, 2, \ldots, e\}$. Each vector has D_2 elements. Note that each feature in the vector represents one word of the $BagOfWords2$ list.

   The process of creating the sequence X' is similar to the process of creating the sequence X. The key difference is that we use the $BagOfWords2$ list of D_2 features instead of the $BagOfWords1$ of D_1 features. Note that, including words that have occurred for at least two times instead of three times into the word list, allows better chance to have more listed words appear in a sentence, which contains a lot fewer words than a segment does.

   Each vector x'_i takes the same label of vector x_i, i.e., $h(x'_i) = h(x_i) = n$, where $i = 1, 2, \ldots, e.$ and $n \in \{1, 2, \ldots, N\}$.

2. Given the sequence of feature vectors, X', and the set of all possible values of labels, the probability of each feature in X' given a label n ($n \in \{1, 2, \ldots, N\}$) is computed using the conditional probability shown in Eq. 6.3:

$$p(j|n) = \frac{Count_j^n + 1}{Count^n + D_2}, \quad j = 1, 2, \ldots, D_2. \tag{6.3}$$

where j represents a feature, $Count_j^n$ represents the count of observed feature j in the vectors that have a label n, $Count^n$ represents the count of all observed features in the vectors that have a label n, and D_2 is the number of features.

Note that we again employ the "add-1" smoothing technique in Eq. 6.3 in order to prevent a zero probability.

3. Each sentence of document C is represented as a D_2-dimension binary feature vector using the word list $BagOfWords2$, where each dimension takes a value of 1 or 0, indicating the presence of the corresponding word in the sentence. Thus, the sentences can be represented as a sequence of T D_2-dimension binary feature vectors, denoted by $O = \{o_i, i = 1, 2, \ldots, T\}$.

Using Eq. 6.3, the computation of the conditional probability of each feature given each possible value of labels (i.e., $n = 1, 2, \ldots, N$) will lead us to compute the initial value of the emission probability of an observation given each state of the HMM, as shown in Eq. 6.4.

$$p(o|n) = \prod_{j=1}^{D_2} p(j|n)^{o^j}, \quad n = 1, 2, \ldots, N, \tag{6.4}$$

where o represents an observation, j represents a feature, o^j represents the value of feature j in observation o, and D_2 is the number of features.

The initial estimated probabilities of $\boldsymbol{\theta}$ will be used in the next subsection for learning the HMM in order to find a best estimation of $\boldsymbol{\theta}$.

6.3.3 Learning the Preliminary HMM

In this subsection, we work on the HMM to learn $\boldsymbol{\theta}$ (i.e., $\boldsymbol{A}$, $\boldsymbol{B}$ and $\boldsymbol{\pi}$) based on Eq. 6.1.

Formally, the HMM, which consists of a sequence of hidden states and independent observations as seen in Figure 2.7, is formed as follows:

Assume that there are T sentences in document C (remember $T = |C|$), denoted by $\{Sen_i, i = 1, 2, \ldots, T\}$, where i represents the position of a sentence in the document (for example Sen_1 and Sen_T denote the first sentence and last sentence of document C, respectively). As shown earlier, the sentences are represented as binary feature vectors to compose a sequence of observations, O.

Each hidden state represents the most likely author of the corresponding sentences. Therefore, there are T hidden states, denoted by $Q = \{q_1, q_2, \ldots, q_T\}$. Each state takes only one possible value from a set denoted by $S = \{1, 2, \ldots, N\}$. For generality, we substitute the set $S = \{1, 2, \ldots, N\}$ by a set $S = \{s_1, s_2, \ldots, s_N\}$.

The estimation of $\boldsymbol{\theta}$, which can explain the observations more effectively, is performed by using the Baum-Welch algorithm (Dempster et al., 1977), which is considered as a special case of the Expectation Maximization (EM) algorithm. The process starts with using the initial values of $\boldsymbol{\theta}$, which were estimated in the previous subsection, and computes the probabilities of being in each state at each time. This is done by using the forward-backward algorithm (Rabiner and Juang, 1986; Bishop, 2006). After that, the estimated probabilities are used to obtain a better estimate of $\boldsymbol{\theta}$. Using the improved (hopefully) $\boldsymbol{\theta}$, the forward-backward algorithm is applied again, and the cycle repeats until the convergence of either the $\boldsymbol{\theta}$ or the estimated probabilities occurs. More details about the Baum-Welch and forward-backward algorithms can be found in Subsection 2.3.3.

The learned $\boldsymbol{\theta}$ will be used in the next subsection in order to find the best sequence of authors that represents the sequence of sentences of document C.

6.3.4 Initial Sentence Decoding

In our problem, we are interested in finding the most likely sequence of states (i.e., authors) that generates the corresponding sequence of observations (i.e., sentences), as shown in Eq. 6.5. Actually, the issue of finding the optimal sequence of states is not the

same as that of finding the individual optimal states. The second issue can be solved by using the forward-backward algorithm to find the state variable marginal and then maximizing each of these separately (Duda et al., 2001).

$$Q^* = \arg\max_{Q} p(Q|O). \tag{6.5}$$

In fact, the number of potential routes through a sequence of states in HMM increases exponentially with the length of the sequence. Therefore, the Viterbi algorithm (Viterbi, 1967; Forney Jr, 1973), also known as max-sum algorithm, is used to find efficiently the most likely sequence of states for the given observations, where the number of potential routes increases only linearly, rather than exponentially, with the length of the sequence. More details about the Viterbi algorithm can be found in Subsection 2.3.4.

After all, by using the Viterbi algorithm, the best sequence of authors, $Q^* = \{q_1, q_2, \ldots, q_T\}$, that represents the corresponding sentences in document C is determined.

6.4 Learning the Boosted HMM

As we have mentioned earlier, the initial values of $\boldsymbol{\theta}$ have a significant impact on the learning process of HMM so that it affects the performance of the decoding process. For unlabelled data, we have proposed a statistical approach to better estimate the initial values of $\boldsymbol{\theta}$ by using segments and learned a preliminary HMM. The HMM has been used to classify each sentence. In this section, the resulted, labelled sentences obtained in the previous section can be used to re-calculate the initial values of $\boldsymbol{\theta}$, which can then be used to learn a more accurate, boosted HMM, to further improve the performance of the decoding.

6.4.1 Creating Consecutive-Sentence Dataset

A procedure, called "Consecutive-Sentence Dataset", is proposed to create a new labelled dataset which can be employed to re-estimate the initial values of $\boldsymbol{\theta}$ and re-construct the HMM. The procedure aims to provide a dataset with a high rate of correctly labelled

data. It strives to provide a dataset with more labelled data for calculating $\boldsymbol{\theta}$, by using sentences rather than segments. This procedure works as follows.

Given the labels of all of the sentences of document C, each sequence of minimum five consecutive sentences that have the same label is inserted into the new dataset with that label.

Eventually, the new dataset, denoted by $CSD = \{(Sentence_1, q'_1), (Sentence_2, q'_2), \ldots, (Sentence_{T'}, q'_{T'})\}$ is created, where $q' = s$ with $s \in S$, and T' represents the number of sentences in CSD.

6.4.2 Re-Estimating and Learning the HMM parameters, and Final-Stage Sentence Decoding

We use the new dataset CSD to re-estimate the new initial values of $\boldsymbol{\theta}$. The computations of the initial values of $\boldsymbol{A}$ and $\boldsymbol{\pi}$ are similar to the computations which have been applied in the previous section, and we replace the set of all labels $h(x_i)$, $i = 1, 2, \ldots, e$ with the set of all states q'_i, $i = 1, 2, \ldots, T'$.

The initial values in $\boldsymbol{B}$ are also re-calculated using the new dataset. However, due to the fact that the new dataset is a sequence of sentences, rather than segments, it is desirable to increase the number of features used in representing the sentences to capture the relation between the observations (i.e., sentences) and the states (i.e., authors). Therefore, a new feature list, denoted by $BagOfWords3 = \{word_1, word_2, \ldots, word_{D_3}\}$, where D_3 is the length of the list, which contains all distinct words that occur at least one time in the document C, is created. By using this list, all sentences in CSD are represented as binary feature vectors, denoted by $X'' = \{x''_i, i = 1, 2, \ldots, T'\}$.

The probability computation of each feature in X'' given a label n ($n \in \{1, 2, \ldots, N\}$) is similar to the computation which has been applied in the previous section. The only difference is that we replace the sequence of vectors, X', of D_2 features by the sequence of vectors, X'', of D_3 features.

Then, the new initial values in $\boldsymbol{\theta}$ (i.e., $\boldsymbol{A}$, $\boldsymbol{\pi}$ and $\boldsymbol{B}$) are utilized for learning the HMM again. The process of learning the HMM is the same as the process discussed in the

previous section. The only difference is that we replace the $BagOfWords2$ list of D_2 features by the $BagOfWords3$ list of D_3 features for representing the observation sequence, O.

Lastly, the final-stage sentence decoding process is applied in order to find the most likely sequence of authors corresponding to all sentences in the document C. Here, the same algorithm illustrated in the previous section is used to perform the decoding process of this step.

Thus far, the SequentialUD approach, which consists of seven steps shown in Figure 6.1, is done.

6.5 Refinement with ModPIP

The works of *Proposed-1* and *Proposed-2*, which are presented in Chapters 3 and 4 respectively, proposed a probability indication procedure (PIP) in order to enhance the purity of sentence classification process. The procedure consists of five criteria. It proceeds by selecting trusted sentences from a document and using them to re-classify each sentence of the document into the author's class. The procedure has been implemented using the Naive-Bayesian model.

Following this idea, in this chapter, a modified version of PIP, named by ModPIP, is proposed to refine the classification results and further improve the sentence classification purity results. Since we treat the sentences of a document as sequential data, the ModPIP is developed based on a sequential model. This is detailed as below.

1. A sentence in the document C, which has been assigned a specific state value, is recorded as a *trusted sentence* if and only if the posterior probability of its state value given the observed sequence of all sentences is greater than the posterior probabilities of all other state values given the observed sequence of all sentences, by more than a threshold R. The state values of the trusted sentences will be fixed.

2. If the first trusted sentence in the document C is not the first sentence in the document, then all sentences starting from the first sentence in the document till the sentence located before the trusted sentence are given the same state value of the trusted sentence.

3. If the last trusted sentence in the document C is not the last sentence in the document, then all sentences starting from the sentence located after the trusted sentence till the last sentence in the document are given the same state value of the trusted sentence.

4. If a group of non-trusted consecutive sentences is surrounded between two trusted sentences that have the same state value, then all the sentences in the group are given the state value of the two trusted sentences.

5. If a group of non-trusted consecutive sentences is surrounded between two trusted sentences that have different state values, then the best split point in the group is picked out in order to divide the group into two subgroups. All the sentences in the first subgroup, which comes before the split point, are given the same state value of the trusted sentence which comes before them. All the sentences in the second subgroup, which comes after the split point, are given the same state value of the trusted sentence which comes after them. The best separation point is the one that gives the maximum summation value of all posterior probabilities of the assigned state values of the sentences in the group given all observed sentences in the document.

The posterior probability of a single state given the observed sequence of all sentences, $p(q|O)$, which is used in the first and fifth criteria, is computed using the forward-backward algorithm.

Regarding Criteria 4 and 5, the number of sentences in a group depends on two factors. The first one is the value of the threshold R that is used to select trusted sentences (see Criterion 1). The second one is the length of a document (i.e., the number of sentences in the document). With the first factor, a small value of threshold R yields a large number of trusted sentences so that the number of sentences in each group is small; and

a large value of R yields a small number of trusted sentences so that the number of non-trusted sentences in each group is comparatively large (see the Experiments section). With the second factor, the document length affects the number of sentences in each group because a short document is supposed to have a smaller number of sentences in each group compared with a long document (see the Experiments section).

6.6 Experiments

In this section, the performance of the proposed approach (i.e., SequentialUD and its refined version) is evaluated and compared with state-of-the-arts on four benchmark datasets widely used for authorship detection. We have used these datasets because the author of each document is known with certainty and because they are canonical datasets that have served as benchmarks for Koppel et al. (2011a), Akiva and Koppel (2012), Akiva and Koppel (2013), Giannella (2015) and Daks and Clark (2016). Furthermore, to test its performance on more realistic cases, randomly selected scientific articles are employed. As an example of a non-artificial document, a scientific paper is also utilized for evaluating the performance of our proposed approach.

6.6.1 Datasets

The three benchmark datasets used for experiments in Chapter 3, i.e., Bible books, Becker-Posner blogs and *New York Times* articles, and an authentic document are also used in this chapter. Furthermore, the single-topic documents of Becker-Posner blogs used for experiments in Chapter 4 are used to evaluate the work of this chapter.

In order to show the efficiency of the proposed approach on more realistic cases, we have randomly selected some scientific articles, which are cited in the Bibliography, covering the same topics. The articles of each topic are mixed in one article and the proposed approach is then applied in order to recover the author of each sentence in the mixed article. Due to the difficulty of finding articles written by single authors covering same topics, as well as due to the fact that in most cases, there is one main author of an article whose writing style can be found throughout of the article, we consider each

article produced by more than one author as an article produced by only one author. In each selected article, we have ignored all metadata (e.g., titles, author names, references, equations, tables and citations). We randomly select two articles on plagiarism detection topic. The articles are Rao et al. (2011) and Kestemont et al. (2011). The lengths of these articles are 66 and 111 sentences, respectively. We also randomly select three articles on authorship attribution topic. The articles are Baayen et al. (2002), Layton et al. (2010) and Savoy (2016). The lengths of these articles are 91, 197 and 304 sentences, respectively. Furthermore, we randomly select four articles on authorship-based text decomposition topic. The articles are Koppel et al. (2011a), Giannella (2015), Daks and Clark (2016) and Aldebei et al. (2016b). The lengths of these articles are 257, 215, 104 and 229 sentences, respectively. The four articles have also used the same data sets in their approaches. Note that, all articles of each topic are randomly selected, in which each article is produced by different authors.

The last corpus tested is the Jane Austen's unfinished novel *Sanditon*. The novel was begun by Jane but interrupted by her death in 1817. In that time, she finished 11 chapters. Many years later, this novel had been completed by "an Other Lady", who had tried to mimic Austen's style and used her notes to finish the novel by writing 19 chapters more. This corpus provides a case to examine our approach in a non-artificial, authentic document.

6.6.2 Experimental Results

The performance of the proposed approach (i.e., SequentialUD and its refined version) is examined through a set of experiments on different documents. In the first four experiments, artificially created documents are created. These documents are created by using the same method that has been used in Chapter 3 (i.e., Section 3.2), which is summarised as follows.

Suppose that there are N authors. Each author has a group of documents. The document of N authors is composed by recursively picking up a random number (m) of unselected successive sentences from a document of a randomly chosen author and merging them together until all sentences in all documents of N authors are selected. During

each iteration, the value of m is randomly chosen from a uniform distribution ranging from 1 to V. We follow the approaches described in Koppel et al. (2011a), Akiva and Koppel (2012) and Akiva and Koppel (2013) and assign 200 to V. In our experiments, we empirically assign 15 to the threshold R for the refined SequentialUD approach. More details can be found in this section.

In order to determine the optimal segment length v, which is employed to estimate the transition matrix A of the preliminary HMM, we group documents from the datasets based on the number of their sentences into two categories, i.e., Long Documents (containing 500 or more sentences), and Short Documents (containing fewer than 500 sentences). Furthermore, based on our observations, the segment length v is also dependent on mean Author Run Length (simplified as meanARL). To depict the impact of document length and meanARL on v, for each category we randomly pick up one document and apply our SequentialUD approach on its sentences with different meanARL. Note that, the meanARL represents the mean number of successive sentences in the document written by the same author. We employ the same procedure described above and use the sentences of the document to create merged documents with different values of meanARL. The meanARL of a document is determined by setting the value of V, which results in a mean of around $0.5V$ successive sentences from the same author on the document. A document resulted from merging the biblical books of Ezekiel and Proverbs (containing 2188 sentences) and the two-student scientific document (containing 313 sentences) have been selected to determine the best segment length for Long and Short Documents, respectively. Tables 6.1 and 6.2 show the purity results of applying our SequentialUD approach in the selected Eze-Prov document (a Long Document) and the two-student scientific document (a Short Document) with different v and meanARL, respectively. As shown in Table 6.1, the proposed approach yields higher purity results in the Eze-Prov document (a Long Document) when v is less than meanARL and 60. Recall that, in our work each author is assumed to have written long successive sentences in a document (i.e., a larger meanARL) and most Long Documents used in our experiments are created with a meanARL of around 100 (i.e., $V = 200$). Since the highest purity result in the Eze-Prov document with the meanARL of around 100 is achieved when v is 30, we assign 30 to v for all Long Documents in our experiments. However,

as shown in Table 6.2, for the scientific document (a Short Document) the proposed approach achieves higher purity results when v is less than meanARL and 40. Also seen from this table, most highest purity results on the scientific document are achieved when v is 10. Therefore, we assign 10 to v for all Short Documents in our experiments. In order to make sure that the number of segments used in the clustering process (Step 3 of estimating transition matrix A) is always larger than the number of clusters, the segment length v for Short Document is set to $\min(10, F(\text{No. Sentences/No. Authors}) - 1)$, where F represents the commonly known floor function.

In our approach, in order to reduce the influence of topics on final results, only those words appearing at least three times in the document are used as features of $BagOfWords1$ to depict the writing style of the segments (see Step 2 of subsection 6.3.2.1). However, note that these words may not necessarily be purely topic-independent words. Based on our observation on different documents, the words selected into the feature set are mostly function words and words that are independent of topics. Increasing the frequency threshold does help to exclude these topic-specific words but at the meantime it also decreases the recall rates of pure segments on the clustering process (Step 3 of the same subsection), and this affects on producing sufficient data for the estimation process. Also note that we use recall rates to evaluate clustering results, because our interest here is to evaluate the capability of the clustering process for retrieving pure segments. Furthermore, we observe that choosing words appearing for at least three times as features of $BagOfWords1$ in a Short or Long Document produces generally higher recall rates on the clustering process. Figure 6.2 shows the recall rates of the clustering process using words that have occurred at least once, twice, three times, four times and five times respectively in four documents as features of $BagOfWords1$. The documents of Becker-Posner Blogs and four-author columnists of New York Times articles have been used as Long Documents. The documents of Traffic Congestion and scientific paper have been used as Short Documents. Obviously, as shown in Figure 6.2, using all words that have occurred three or more times has achieved higher recall rates on the clustering process for all four documents.

TABLE 6.1: Purity results of applying our SequentialUD approach on the selected Eze-Prov document (a Long Document) with different v and meanARL. Note that better purity results (highlighted in bold font) are achieved when v is less than meanARL and 60.

		Segment Length v									
		5	**10**	**20**	**30**	**40**	**50**	**60**	**70**	**80**	**90**
meanARL	**10**	**85.5%**	75.4%	61.4%	61.1%	59.2%	57.7%	60.3%	55.3%	56.2%	57.4%
	20	**91.9%**	**92.4%**	57.3%	60.1%	55.1%	58.2%	55.3%	55.3%	56.4%	56.1%
	30	**93.8%**	**94.2%**	**95.2%**	59.6%	58.1%	60.1%	57.5%	56.0%	57.3%	57.1%
	40	**93.7%**	**93.8%**	**94.3%**	**94.2%**	55.6%	55.5%	52.6%	55.7%	60.6%	58.3%
	50	**90.4%**	**91.6%**	**96.9%**	**96.4%**	**94.0%**	56.0%	55.6%	60.2%	59.8%	57.3%
	60	**90.2%**	**92.5%**	**96.3%**	**97.0%**	**94.2%**	**83.1%**	55.3%	55.7%	56.2%	56.0%
	70	**89.6%**	**96.7%**	**97.3%**	**97.2%**	**93.4%**	**88.5%**	68.1%	60.9%	60.4%	58.4%
	80	**90.0%**	**94.6%**	**97.9%**	**98.2%**	**98.0%**	**90.1%**	63.1%	60.6%	61.7%	59.1%
	90	**88.4%**	**97.7%**	**98.6%**	**99.0%**	**97.7%**	**97.0%**	85.6%	61.7%	57.6%	55.0%
	100	**87.0%**	**98.0%**	**98.1%**	**99.2%**	**95.6%**	**96.2%**	84.8%	62.7%	63.6%	62.8%

TABLE 6.2: Purity results of applying our SequentialUD approach on the scientific document (a Short Document) with different v and meanARL. Note that better purity results (highlighted in bold font) are achieved when v is less than meanARL and 40.

		Segment Length v									
		5	**10**	**20**	**30**	**40**	**50**	**60**	**70**	**80**	**90**
meanARL	**10**	**76.4%**	66.8%	57.8%	51.8%	50.5%	53.7%	57.5%	52.1%	51.1%	51.1%
	20	**85.0%**	**86.6%**	78.6%	52.7%	56.2%	62.3%	50.8%	51.1%	52.4%	54.0%
	30	**85.6%**	**89.1%**	**84.0%**	54.6%	50.2%	51.4%	52.7%	52.7%	53.4%	53.0%
	40	**87.2%**	**93.0%**	**91.7%**	**90.7%**	58.8%	61.0%	60.7%	54.0%	55.6%	57.8%
	50	**86.6%**	**91.7%**	**89.1%**	**88.2%**	74.1%	60.4%	56.9%	54.3%	55.6%	55.6%
	60	**89.5%**	**91.4%**	**90.1%**	**85.9%**	78.3%	64.2%	54.0%	55.3%	60.4%	55.3%
	70	**89.8%**	**92.7%**	**91.7%**	**89.8%**	82.1%	77.7%	73.5%	70.3%	58.8%	54.3%
	80	**91.1%**	**93.6%**	**93.6%**	**92.7%**	88.2%	88.5%	74.7%	60.8%	58.5%	57.2%
	90	**88.2%**	**94.6%**	**94.9%**	**92.0%**	80.2%	84.0%	73.2%	54.0%	54.3%	54.0%
	100	**88.5%**	**95.8%**	**94.3%**	**91.7%**	86.3%	82.1%	77.3%	71.6%	73.8%	59.4%

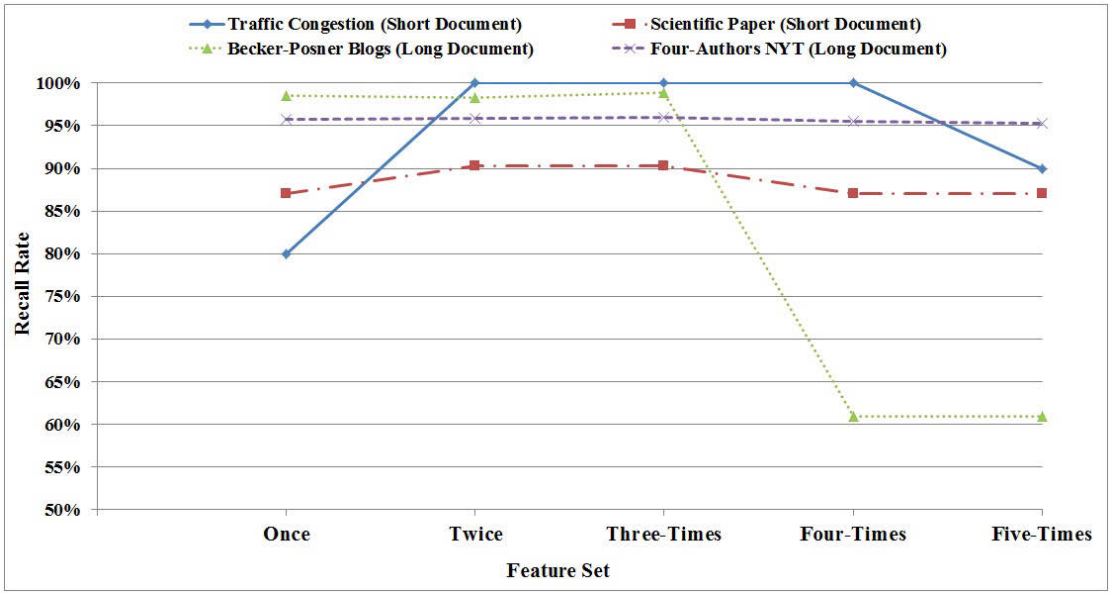

FIGURE 6.2: The recall rates of the clustering process obtained using words that have occurred at least once, twice, three times, four times and five times in four documents as features of BagOfWords1.

The results of the proposed approach SequentialUD and its refined version are compared with those obtained by five state-of-the-art approaches.

6.6.2.1 Results on the Biblical Books Dataset

For the first set of experiments, the five biblical books of five authors are utilized to produce a set of total 10 merged documents of two authors by using the procedure mentioned before. The 10 documents are related to either different genres or the same genre (see Table 3.7). In Tables 6.3 and 6.4, we report the purity results obtained by applying the proposed approach SequentialUD and its refined version on the documents composed by merging two biblical books of different genres (Table 6.3) and the documents composed by merging two biblical books of the same genre (Table 6.4), respectively. In both cases, the results of the SequentialUD and its refined version are compared with the approaches of Koppel et al. (2011a), Akiva and Koppel (2013), Akiva and Koppel (2013)-SynonymSet and *Proposed-1*. The purity results of the approach in Akiva and Koppel (2012) are used further for comparison in Table 6.4.

TABLE 6.3: Purity comparison on documents composed by merging two biblical books of *different genres*. Approaches in comparison: 1- Koppel et al. (2011a), 2- Akiva and Koppel (2013), 3- Akiva and Koppel (2013)-SynonymSet, 4- *Proposed-1*, 5- Our SequentialUD and 6- Our refined SequentialUD.

Doc.	**1**	**2**	**3**	**4**	**5**	**6**
Jer-Prov	72.7%	97.0%	75.0%	99.0%	99.6%	99.8%
Isa-Job	82.2%	98.7%	89.1%	98.7%	99.6%	99.6%
Eze-Prov	76.6%	98.7%	90.8%	97.9%	99.2%	99.4%
Isa-Prov	70.4%	95.0%	85.0%	97.9%	98.7%	99.2%
Eze-Job	85.9%	98.7%	95.0%	99.0%	99.7%	99.7%
Jer-Job	87.3%	98.0%	93.1%	97.8%	99.1%	99.2%
Overall	***79.2%***	***97.7%***	***88.0%***	***98.4%***	***99.3%***	***99.5%***

TABLE 6.4: Purity comparison on documents composed by merging two biblical books of the *same genre*. Approaches in comparison: 1- Koppel et al. (2011a), 2- Akiva and Koppel (2012), 3- Akiva and Koppel (2013), 4- Akiva and Koppel (2013)-SynonymSet, 5- *Proposed-1*, 6- Our SequentialUD and 7- Our refined SequentialUD.

Doc.	**1**	**2**	**3**	**4**	**5**	**6**	**7**
Job-Prov	84.5%	84.9%	93.9%	82.0%	95.2%	98.6%	99.2%
Jer-Eze	82.0%	87.6%	96.6%	95.9%	97.0%	97.7%	98.2%
Isa-Jer	71.8%	63.4%	66.7%	82.7%	71.0%	73.1%	73.3%
Isa-Eze	78.9%	76.0%	80.0%	88.0%	82.7%	83.6%	83.8%
Overall	***79.3%***	***78.0%***	***84.3%***	***87.2%***	***86.5%***	***88.3%***	***88.6%***

From the purity results presented in the tables, we can observe that the results obtained with our proposed approach SequentialUD and its refined version are quite promising with a purity of over 99.5% achieved on some documents. We can also see that the overall purities of our proposed approach are remarkably better than those obtained using other approaches. In some cases (e.g., for the Jer-Prov document mentioned in Table 6.3), SequentialUD produces a 37% larger purity result than Koppel et al. (2011a) and 33% larger purity result than Akiva and Koppel (2013)-SynonymSet. Note that, two of them, i.e., Koppel et al., 2011 and Akiva and Koppel, 2013-SynonymSet, are specially developed for biblical books only, and not applicable for other documents.

6.6.2.2 Results on Becker-Posner Blogs Dataset (Controlling for Topic)

For the second set of experiments, we apply the proposed approach on the merged documents composed from the Becker-Posner blogs corpus.

On the first part of our experiments using this corpus, we work on a document created by merging all Becker blogs and Posner blogs. The merged document has 26,922 sentences and 246 turns between the two authors. It does not have any topic indication that can be used to differentiate between authors. As shown in Table 6.5, the purity results achieved by applying our proposed approach (i.e., SequentialUD and its refined version) on this document are significantly higher.

In fact, it is important to know the effectiveness of applying the procedures of our SequentialUD approach. These procedures are the preliminary HMM, the Boosted HMM and the ModPIP refinement. Table 6.5 shows the intermediary and final purity results achieved by applying our SequentialUD approach on Becker-Posner blogs. In this table, "4-First-Stage HMM" is the purity obtained by applying the first-state preliminary HMM, and "5-Our SequentialUD" is the purity obtained after further applying the Boosted HMM, and "6-Our Refined SequentialUD" is the result obtained after applying the ModPIP refinement in the end. From these results, it can be seen clearly that: 1) The purity achieved using our preliminary HMM is already very effective and has outperformed the other three approaches; 2) Our BoostedHMM and ModPIP refinement have further improved the purity results, each by 0.6%.

TABLE 6.5: Purity comparison on a document of Becker-Posner Blogs. Approaches compared: 1- Akiva and Koppel (2012), 2- Akiva and Koppel (2013), 3- *Proposed-1*, 4- First-Stage HMM, 5-Our SequentialUD and 6- Our Refined SequentialUD.

Document	**1**	**2**	**3**	**4**	**5**	**6**
Becker-Posner Blogs	94.0%	94.9%	96.6%	96.7%	97.3%	97.9%

As we have mentioned earlier (see Section 6.4), we have used each sequence of minimum five consecutive sentences that have the same label to create the consecutive-sentence dataset. In fact, the limit of five sentences depends on a mean author run length (i.e., the mean of the numbers of consecutive sentences from the same author in a document).

Figure 6.3 presents an example of purity results achieved on Becker-Posner blogs when the SequentialUD and its refined version are applied using different values of the limitation. Clearly, it can be seen that the purity results are not very sensitive to the value of this setting (i.e., five consecutive sentences) as long as the value does not exceed the mean author length in the document. In this chapter, we set the value of the limitation to five because no document tested in our experiments has a mean author run length less than five.

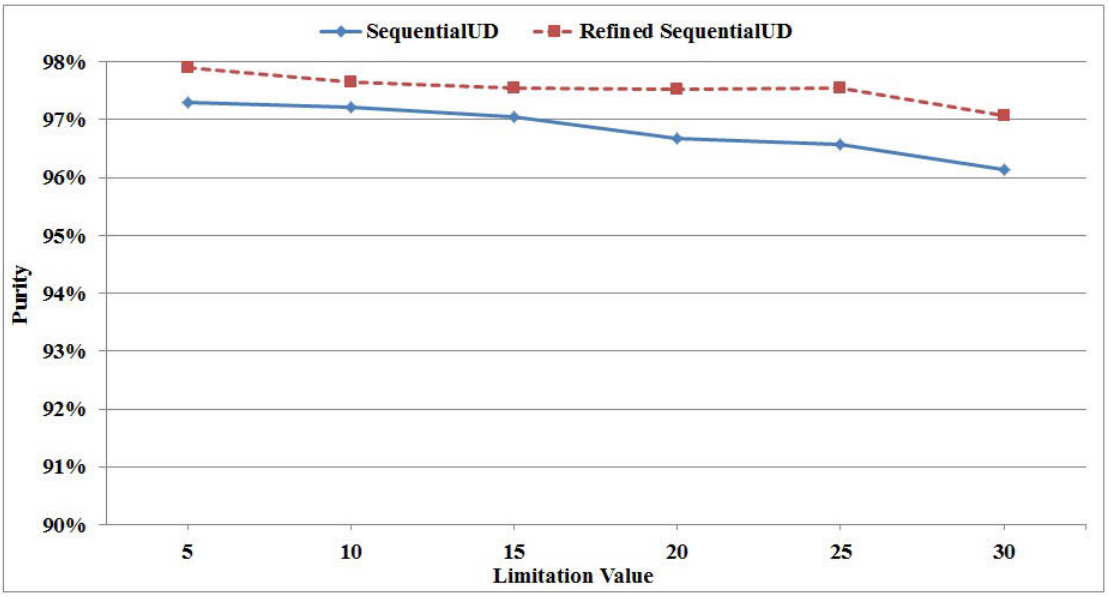

FIGURE 6.3: Purity results achieved on Becker-Posner blogs when our SequentialUD and its refined version are applied using different values of the limitation used to create the consecutive-sentence dataset.

On the second part of our experiments regarding this corpus, the six single-topic documents (see Table 4.1) manually created by Giannella (2015) from the Becker-Posner blogs are used to test the performance of the proposed approach, where each document has sentences representing only one single topic. Figure 6.4 illustrates the purity results obtained using our proposed SequentialUD approach and its refined version, compared with that of the approach in Giannella (2015). As shown in the figure, both versions of our approach have yielded better purity results (up to 42.5% in the "Traffic Congestion (TC)" document) in all six documents.

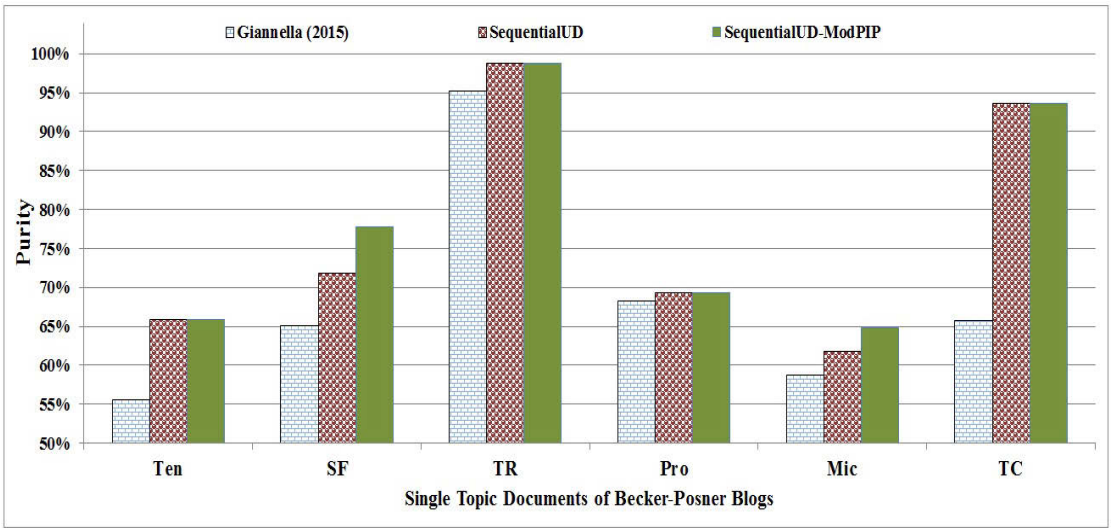

FIGURE 6.4: Purity results of the approaches proposed by Giannella (2015), our SequentialUD and our refined SequentialUD using the six single-topic documents of Becker-Posner blogs.

For the rest of this subsection, we would like to show the impact of the V value on performance. As we have mentioned earlier, our experiments are run on artificially merged documents (excluding the six single-topic documents of the Becker-Posner blogs). For each document, the number of consecutive sentences of each author before turning to another author is randomly chosen from a uniform distribution ranging from 1 to V (see Section 3.2), i.e., a mean of around $0.5V$ sentences between transitions among authors is expected. Smaller values of V produce relatively larger numbers of turns between authors, and make the document decomposition become harder. Therefore, the range of the uniform distribution (i.e., the value of V) somehow determines the complexity of the problem. To demonstrate how susceptible our results are to the complexity of document decomposition problem, Figure 6.5 presents the purity results obtained by using the merged documents of Becker-Posner blogs created by assigning different values of V using our proposed approach SequentialUD and its refined version. Clearly, for small values of V, the purity results of the proposed approach are somewhat degraded. In our experiments, we have set $V = 200$.

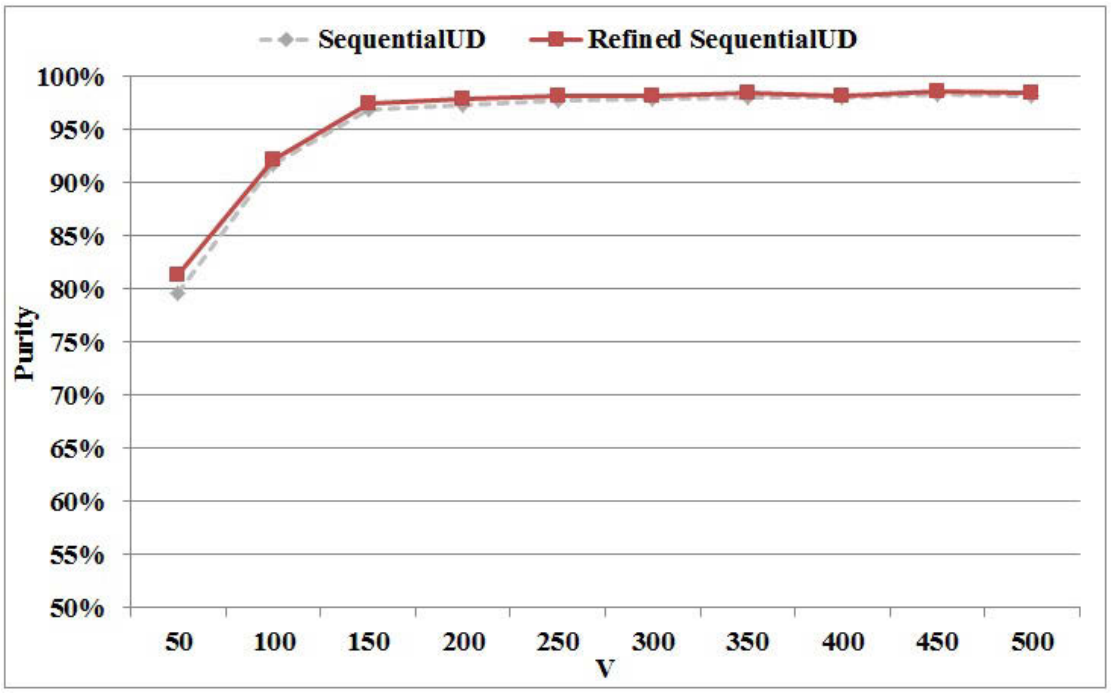

FIGURE 6.5: Purity results obtained by using the merged documents of Becker-Posner blogs created by assigning different values of V using our proposed approach SequentialUD and its refined version.

6.6.2.3 Results on *New York Times* Articles Dataset (N $\geq$ 2)

In these experiments, we employ the *New York Times* articles of four columnists to create a set of merged documents, and the merged documents have two, three or four authors.

In the first set of experiments regarding this corpus, all possible documents of two authors are composed and six documents are produced. Table 6.6 lists the six resulted documents and the number of turns between authors in each document. Table 6.6 also displays the purity results obtained using our proposed approach, i.e. SequentialUD and its refined version, and the *Proposed-1* approach of the six documents.

From Table 6.6, it can be seen that the purity results of our approach have exceeded the results obtained using the *Proposed-1* in all of the six documents. These have also outperformed the purity of 88.0%, acquired by Akiva and Koppel (2012) and Akiva and Koppel (2013).

TABLE 6.6: The purity results of documents created by merging any pair of the four *New York Times* columnists using the *Proposed-1* approach, our SequentialUD and our refined SequentialUD.

	Doc.	No. of Turns	Proposed-1	SequentialUD	Refined SequentialUD
1	TF-PK	251	95.6%	95.8%	96.3%
2	GC-PK	253	93.7%	95.0%	96.6%
3	GC-TF	242	96.1%	96.8%	98.0%
4	MD-PK	255	95.5%	97.0%	98.2%
5	MD-TF	249	93.3%	94.1%	96.0%
6	MD-GC	251	93.8%	94.7%	95.5%

In the second set of our experiments regarding this corpus, all possible documents of three or four authors are composed. This results in four documents of three authors and one document of four authors. Each document composed by three authors has on average more than 350 turns between the authors. The document composed by four authors has more than 500 turns between the authors. Figure 6.6 shows the purity results of applying our SequentialUD and refined SequentialUD on the five aforementioned documents (i.e., four documents having three authors and one document having four authors), comparing with the approaches of Akiva and Koppel (2012), Akiva and Koppel (2013) and *Proposed-1*.

As shown in Figure 6.6, the purities achieved by our SequentialUD approach and its refined version are significantly higher no matter if a document is written by three or four authors. In addition, it should be noted that the proposed approach consistently outperforms the other three state-of-art approaches in all of the five documents. Figure 6.6 also shows that, in the experiments involving the four-author document (i.e., MD-GC-TF-PK), the refined SequentialUD produces a 34% higher purity than the approach in Akiva and Koppel (2013). Once again, comparing the purity results obtained in all of the five documents using the refined and non-refined version of our approach, one can see that, applying our ModPIP on the BoostedHMM has further improved the performance by 2.2% on average. This clearly demonstrates the effectiveness of the ModPIP procedure.

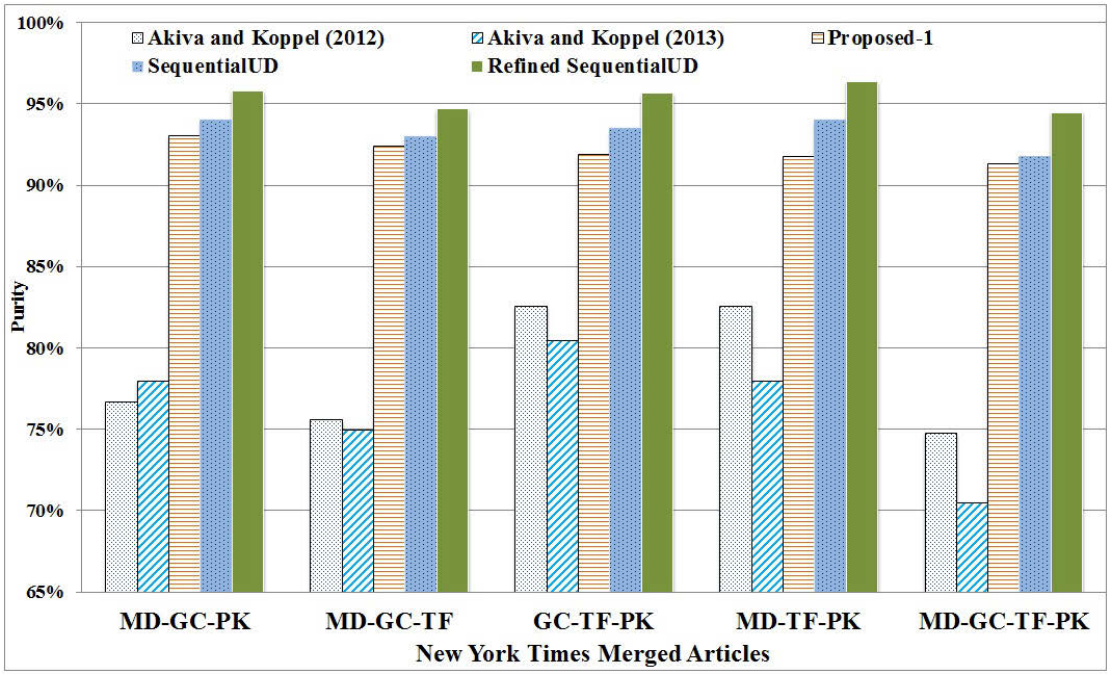

FIGURE 6.6: Purity results of the approaches proposed by Akiva and Koppel (2012), Akiva and Koppel (2013), *Proposed-1*, SequentialUD and refined SequentialUD using documents composed by merging articles of three and four *New York Times* columnists.

It would be interesting to know the impact of the threshold, R, which is used for Criterion 1 of the ModPIP, on the purity results and on the possible number of sentences that could be contained in a group of non-trusted consecutive sentences for Criteria 4 and 5 of the ModPIP. Figure 6.7 presents the purity results of refined SequentialUD approach using the document written by four *New York Times* columnists when different values of threshold R are used in the ModPIP. The figure also shows the maximum number of sentences that are located in a group regarding Criteria 4 and 5 of the ModPIP when different values of threshold R are used in this document.

As it can be seen that, when the value of threshold R is small, the maximum number of sentences in a group is small and the purity results are not high enough. The reason of that is because a small value of R yields a large number of trusted sentences, with less confidence in the correctness of their labels, and so the numbers of sentences in groups are small. The less confidence in the labels of the resulted trusted sentences directly affects the correctness of labels of sentences in the groups and so the purity result of a whole document is relatively low. On the other side, when the value of threshold R is large, the maximum number of sentences in a group is comparatively large and

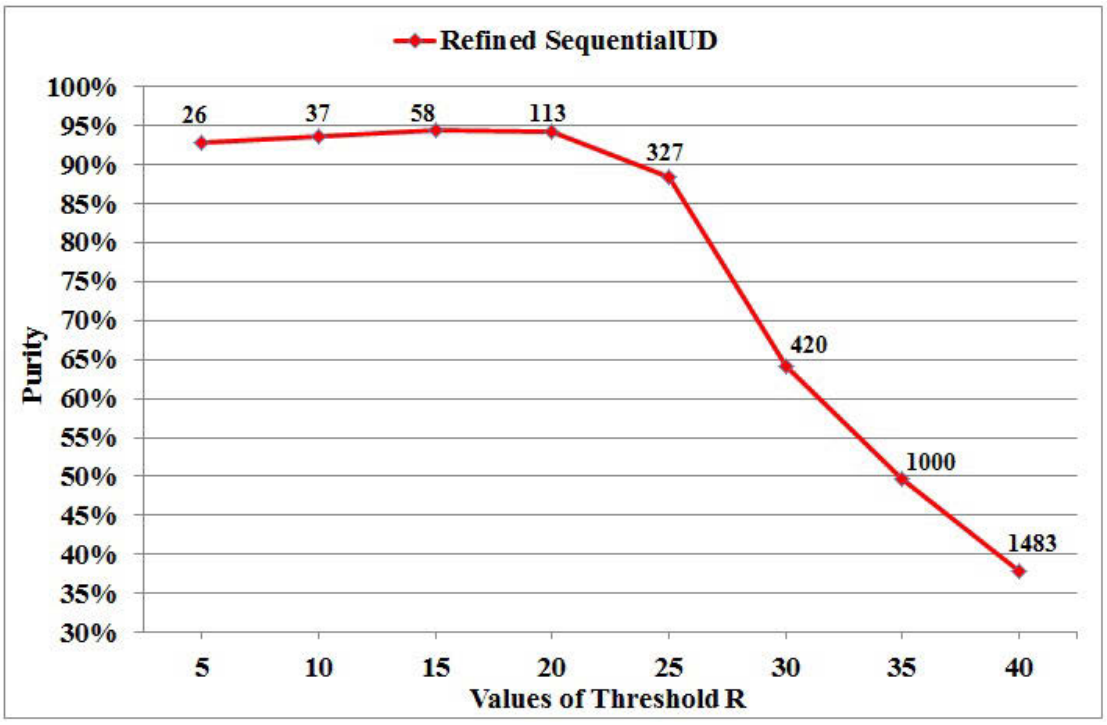

FIGURE 6.7: Purity results of the refined SequentialUD approach with respect to the maximum number of sentences located in a group, as indicated in Criteria 4 and 5 of the ModPIP, using the document written by four *New York Times* columnists when different values of threshold R (the horizontal axis) are used in the ModPIP. In the graph, the numbers above the line markers indicate the maximum number of sentences located in a group.

the purity results drop down sharply. The reason of that is because a large value of R yields a small number of trusted sentences and so the number of sentences in groups are large. All sentences contained in a group and surrounded by two far trusted sentences have a very low chance to have correct labels which are assigned depending on the two trusted sentences and so the purity results of a whole document is low. Therefore, in the case of having a very long sequence of non-trusted sentences surrounded by two trusted sentences, a smaller value of threshold R should be selected.

In order to examine the proposed approach in shorter documents, another set of experiments regarding the *New York Times* corpus is applied. In this set of experiments, merged documents of two, three and four columnists composed of only n different randomly selected articles of each columnist are created. For each resulted merged document, the SequentialUD approach and its refined version are applied and the purity results are computed. We repeat this process 50 times and then the mean purity results

over the 50 trials for SequentialUD and its refined are computed. The 0.95 confidence intervals over the 50 purity results for the refined SequentialUD version are also computed. Figure 6.9 shows the purity results of SequentialUD approach and its refined version on merged documents created by merging 1, 5, 10, 15 and 20 randomly selected articles of two, three and four authors. Figure 6.9 also shows the 0.95 confidence intervals for the refined SequentialUD version.

As shown in Figure 6.9, the purity results obtained with our proposed approach SequentialUD and its refined version on *New York Times* short documents are quite promising.

In Giannella (2015), the author has examined his approach of document decomposition on short documents created by merging a few sentences of the four columnists of New York Times articles. These documents are created using a procedure which is different from the one used in this article. The procedure aims to create a merged document containing a specific number of runs of successive sentences of each columnist. Giannella has performed the procedure for 100 trials. In each trial, a multi-author document is created, his approach on document decomposition is performed, and a matching accuracy is then computed. After that, the mean and the 0.95 confidence intervals over the 100 accuracies of the approach are computed. The experiments applied in this article (excluding the six single topic documents) have assumed that there is a long sequence of consecutive sentences for each author. It is interesting to see how our proposed approach can be applied in short documents with short consecutive sentences. Therefore, we create short documents of four columnists of *New York Times* articles using the same procedure of Giannella (2015). Each created merged document contains exactly two runs of each columnist (i.e., there are seven transitions form one author to another). During each run of each columnist, the number of selected successive sentences of the columnist in that run is randomly chosen from an exponential distribution with meanARL (when the chosen number is not an integer, we round it to the nearest integer). Note that, the meanARL determines the mean number of successive sentences from the same author on the merged document. Figure 6.8 shows the purity results of SequentialUD approach and its refined version in short documents, which are composed by merging articles of four New York Times columnists using the same procedure of Giannella (2015), when the mean author run length (i.e., meanARL) is varied.

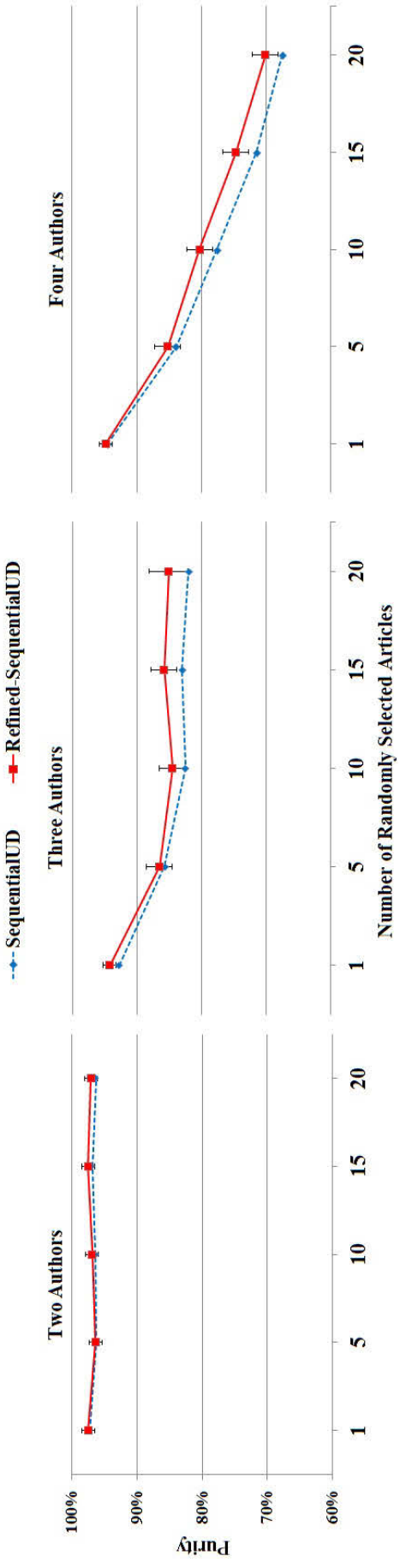

FIGURE 6.9: Purity results of SequentialUD approach and its refined version on merged documents created by merging 1, 5, 10, 15 and 20 randomly selected articles of two, three and four authors. The error bars depict 0.95 confidence interval for the refined SequentialUD approach. In many cases the confidence intervals are quite small and are not easily seen in the figure.

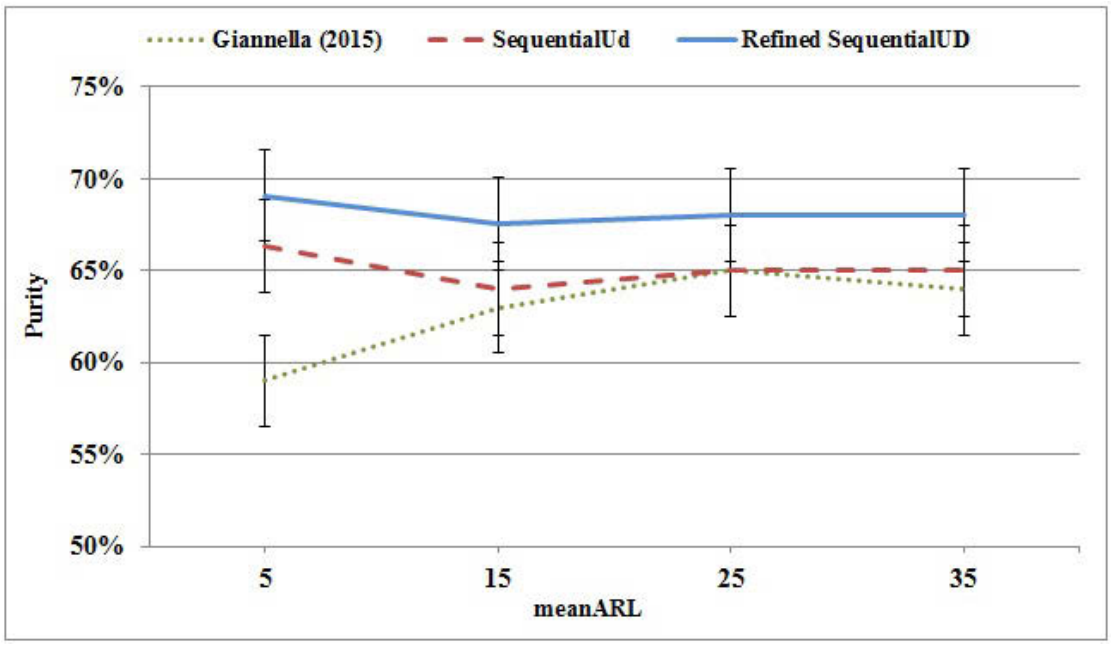

FIGURE 6.8: Comparison of the purity results obtained using the approach in Giannella (2015), SequentialUD approach and refined SequnatialUD approach on short documents with short consecutive sentences composed by merging articles of four *New York Times* columnists using the same procedure of Giannella (2015), when the mean author run length (i.e., meanARL) is varied. The error bars depict 0.95 confidence interval for the three approaches.

As shown in Figure 6.8, the purity results of the SequentialUD and its refined version outperform the results of Giannella's approach in all values of meanARL (except in the case of meanARL = 25 where SequentialUD achieves a purity equal to that achieved in Giannella's approach).

6.6.2.4 Results on Randomly Selected Scientific Articles

To show the efficiency of our SequentialUD and its refined version on more realistic cases, we employ randomly selected scientific articles covering the same topics and create a set of merged articles. Each merged article has two, three or four authors and the topics among authors are not differentiated. In the first set of experiments regarding the scientific articles, we create a merged article using two randomly selected scientific articles on plagiarism detection topic (i.e., Rao et al. (2011) and Kestemont et al. (2011)). The

merged article consists of 177 sentences written by two authors. Our proposed approach classifies sentences of the two authors with 94.9% by using SequentialUD and 98.3% by using its refined. In the second set of experiments regarding the scientific articles, we create a merged article using three randomly selected scientific articles on authorship attribution topic (i.e., Baayen et al. (2002), Layton et al. (2010) and Savoy (2016)). The merged article consists of 592 sentences written by three authors. Our proposed approach classifies sentences of the three authors with 92.2% by using SequentialUD and 92.4% by using its refined. In the last set of experiments regarding the scientific articles, we create a merged article using four randomly selected scientific articles on authorship-based document decomposition topic (i.e., Akiva and Koppel (2013), Giannella (2015), Daks and Clark (2016) and Aldebei et al. (2016b)). The merged article consists of 805 sentences written by four authors. Our proposed approach classifies sentences of the four authors with 93.7% by using SequentialUD and 98.8% by using its refined. It is clear that the purity results obtained with our approach on merged articles of two, three or four authors are quite promising.

6.6.2.5 Results on *Sanditon*: An Unfinished Novel

To show that our approach also works well on non-artificial, authentic documents, we apply our SequentialUD and its refined version on the *Sanditon* novel. The novel, which has been written by Jane Austen and an unknown lady, contains 30 chapters. Jane wrote 11 chapters and the lady wrote 19 chapters. Each chapter contains on average 164 sentences. Our proposed approach classifies Austen's sentences from Another Unknown Lady with 93.3% by using SequentialUD and 95.2% by using its refined. It is clear that the proposed approach has achieved promising purity results. In a case of assuming that we know that each chapter of the novel is written by only one author and assign each chapter to the author who has most of the sentences of that chapter, we achieve purity results equal to 96.7% by applying the SequentialUD and its refined version (one mislabelling chapter). Furthermore, when we compare our results with that acquired by Daks and Clark (2016), a purity result of 93.8% is reported by using this approach. It is worth to note that the approach of Daks and Clark (2016) aims to cluster chapters, not sentences, assuming that each whole chapter is written by one author.

6.6.2.6 Results on Scientific Document

To also work on other types of non-artificial, scientific document, we have applied our SequentialUD approach and its refined version on a scientific paper initially drafted by two Ph.D students (i.e., Students A and B). Each student has written three sections. Students A and B have contributions on the paper equal to 41.9% and 58.1%, respectively. Table 6.7 presents the number of correctly classified sentences of each author and the purity of the classification process using our SequentialUD approach and its refined version. Furthermore, Table 6.7 shows the predicted contributions of each author using the proposed approach.

As shown in Table 6.7, the purity results and the predicted contributions for each author using the proposed approach on a non-artificial, scientific document are very promising.

The results of the scientific document are obtained when information that each author of the paper has written a whole section is not available. In a case of assuming that we know that each whole section of the scientific paper is written by only one author and assign each section to the author who has most of the sentences of that section, we achieve a purity results equal to 97.4% by applying the SequentialUD and its refined version.

When we have applied the refined SequentialUD approach on the scientific document, we investigate the number of sentences that are located in groups between two trusted sentences (i.e., Criteria 4 and 5 of the ModPIP) and we find that 10 is the maximum number of sentences that are located in the groups. Table 6.8 shows the maximum number of sentences that are located in groups regarding Criteria 4 and 5 of the ModPIP using all corpus used in this article when the value of threshold R is equal to 15. Table 6.8 also shows that the maximum number of sentences in all groups in each long document (having more than or equal 500 sentences) and the maximum number of sentences in all groups in each short document (having fewer than 500 sentences) are ranging from 30-58 sentences and 10-12 sentences, respectively.

TABLE 6.7: The purity results and predicted contributions of the two authors of the scientific paper using the proposed approach SequentialUD and its refined version.

		SequentialUD Approach			**Refined SequentialUD Approach**		
Author	**No. of Sentences**	**Correctly Classified Sentences**	**Purity**	**Contribution**	**Correctly Classified Sentences**	**Purity**	**Contribution**
A	131	131	100%	46.3%	131	100%	45.0%
B	182	168	92.3%	53.7%	172	94.5%	55.0%
Overall Purity		**95.5%**			**96.8%**		

TABLE 6.8: The maximum number of sentences that are located in groups regarding criteria 4 and 5 of the ModPIP using corpus used in this article when the value of threshold R is equal to 15.

Corpus Name	**Maximum Number of Sentences in Groups**
Bible Books	30 Sentences
Becker-Posner Blogs	34 Sentences
Becker-Posner Blogs (Single Topic)	12 Sentences
New York Times Articles (2 Authors)	41 Sentences
New York Times Articles (3 Authors)	57 Sentences
New York Times Articles (4 Authors)	58 Sentences
Sanditon	35 Sentences
Scientific Paper	10 Sentences

6.7 Summary

In this chapter, aiming at segmenting a multi-author document into components according to authorship, we have proposed to utilise the useful sequential correlation among the consecutive sentences in order to determine the authorial components. The proposed approach is based on the well-known sequential model, i.e., Hidden Markov Model, in order to find the best sequence of authors that represents the corresponding sequence of sentences in the document. Our previously proposed probability indication procedure, which has been presented in Chapters 3 and 4, has been further modified and used to further improve the purity results by considering sequential patterns.

The experimental results have shown that our proposed approach has achieved high purity results on all the datasets used in this chapter and have obviously outperformed the approaches proposed by Koppel et al. (2011a), Akiva and Koppel (2012), Akiva and Koppel (2013), Giannella (2015), Daks and Clark (2016) and *Proposed-1* in terms of purity.

Chapter 7

Conclusions

This PhD thesis has presented new approaches for the authorship-based multi-author document decomposition. The main findings of this work are recapitulated as follows.

- In Chapter 3, we have proposed an approach for decomposing a multi-author document into authorial components. The approach aims to avail the differences of the posterior probabilities of the Naive Bayesian model in order to enhance the performance of the sentence segmenting process. It has been shown that using the proposed segment elicitation procedure can be beneficial to pick out from a cluster only the strongest and most effective segments that can best represent the writing style of the cluster to be used in training a supervised Naive Bayesian classifier for segmenting all sentences in the document. It has also been shown that using the proposed probability indication procedure can greatly increase the performance of the sentence decomposing process. The main idea of the procedure is to select the significant and trustful sentences from a document and involve them to re-decompose all sentences in the document. The proposed approach has been evaluated on three benchmark datasets, of which every one has its own characteristics, and has obtained significantly high purity results.

- In Chapter 4, we have extended the proposed approach presented in Chapter 3 and proposed an unsupervised, hierarchical learning framework for authorship-based multi-author document decomposition. The main idea of the proposed approach

is to produce a more powerful training dataset, better than the one used to train the classifier in our previously approach presented in Chapter 3, with more accurate labels to improve the classifier's robustness and performance. Experimental results of the proposed approach over three benchmark datasets, including single-topic documents, have reported interesting performance in decomposing sentences of short and long documents into authorial components. A scientific paper has also been tested and used to show the effectiveness of the proposed approach on authentic document.

- In Chapter 5, we have proposed another approach for authorship-based multi-author document decomposition. The approach is based on utilizing the sequential pattern hidden among sentences for determining their authorships. The well-known sequential model, i.e., Hidden Markov Model, has been adopted to find the best sequence of authors that represents the corresponding sequence of sentences in the document. It has been shown that these sequential patterns can be surprisingly useful in decomposing the sentences into authorial components. A new unsupervised method for estimating the initial values of the HMM has also been proposed in this chapter. It has been noticed, as shown in Figure 5.1, that using this method has really increased the decomposing purity results. The proposed approach, unlike the approaches presented in Chapters 3 and 4, does not require estimation of any threshold. In this chapter, as shown in Section 5.6.3, we have also mentioned an application (i.e., authorship attribution) where the multi-author decomposition can be applied.

- In Chapter 6, we have proposed to utilise the useful sequential correlation among the consecutive sentences in order to determine the authorial components. A Hidden Markov Model (HMM) is also constructed to explore the sequential correlation in a document. Our comparative evaluation results with the state-of-the-arts have demonstrated the strength of our proposed idea in terms of effectively decomposing the document into sentences according to their authorship, regardless of their topics, languages, etc. It has been noticed that the performance of the proposed approach is better when the length of successive sentences of each author is relatively long. The great strength of our refined SequentialUD approach is

the selection of the trusted sentences from the document and using them in re-classifying sentences, such as very short sentences, that do not have sufficiently discriminative features. However, the proposed approach may not yet be effective to predict the authors of sentences when the majority of the sentences are very short. For example, each message in a chat on a social network (e.g., *Facebook*) often contains a few words only.

Some extensions of this work could be undertaken in the immediate future. For example, an automatic approach for determining the number of authors of a multi-author document could be proposed. Furthermore, an adaptive learning method to select the optimal values of the thresholds used in the approaches presented in Chapters 3 and 4 could be explored.

Bibliography

Navot Akiva and Moshe Koppel. Identifying distinct components of a multi-author document. In *EISIC*, pages 205–209, 2012.

Navot Akiva and Moshe Koppel. A generic unsupervised method for decomposing multi-author documents. *Journal of the American Society for Information Science and Technology*, 64(11):2256–2264, 2013.

Chris Giannella. An improved algorithm for unsupervised decomposition of a multi-author document. *Journal of the Association for Information Science and Technology*, 2015.

Moshe Koppel, Navot Akiva, Idan Dershowitz, and Nachum Dershowitz. Unsupervised decomposition of a document into authorial components. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 1356–1364. Association for Computational Linguistics, 2011a.

Khaled Aldebei, Xiangjian He, and Jie Yang. Unsupervised Decomposition of a Multi-Author Document Based on Naive-Bayesian Model. *Association for Computational Linguistics, Volume 2: Short Papers*, page 501, 2015.

Khaled Aldebei, Xiangjian He, Wenjing Jia, and Jie Yang. Unsupervised Multi-Author Document Decomposition Based on Hidden Markov Model. In *ACL (1)*, 2016a.

Khaled Aldebei, Helia Farhood, Wenjing Jia, Priyadarsi Nanda, and Xiangjian He. Sequential and Unsupervised Document Authorial Clustering Based on Hidden Markov Model. In *Trustcom/BigDataSE/ICESS, 2017 IEEE*, pages 379–385. IEEE, 2017.

Khaled Aldebei, Xiangjian He, Wenjing Jia, and Weichang Yeh. SUDMAD: Sequential and Unsupervised Decomposition of a Multi-Author Document Based on a Hidden Markov Model. *Journal of the Association for Information Science and Technology*, 69(2):201–214, 2018. ISSN 2330-1643. doi: 10.1002/asi.23956. URL http://dx.doi.org/10.1002/asi.23956.

Thorsten Brants, Francine Chen, and Ioannis Tsochantaridis. Topic-based document segmentation with probabilistic latent semantic analysis. In *Proceedings of the eleventh international conference on Information and knowledge management*, pages 211–218. ACM, 2002.

Leonhard Hennig and DAI Labor. Topic-based multi-document summarization with probabilistic latent semantic analysis. In *RANLP*, pages 144–149, 2009.

Pedro Mota, Maxine Eskenazi, and Luísa Coheur. Multi-document topic segmentation using bayesian estimation. In *Semantic Computing (ICSC), 2016 IEEE Tenth International Conference on*, pages 443–447. IEEE, 2016.

Francesca Cesarini, Marco Gori, Simone Marinai, and Giovanni Soda. Structured document segmentation and representation by the modified xy tree. In *Document Analysis and Recognition, 1999. ICDAR'99. Proceedings of the Fifth International Conference on*, pages 563–566. IEEE, 1999.

Pınar Duygulu and Volkan Atalay. A hierarchical representation of form documents for identification and retrieval. *International Journal on Document Analysis and Recognition*, 5(1):17–27, 2002.

Alon Daks and Aidan Clark. Unsupervised authorial clustering based on syntactic structure. *ACL 2016*, page 114, 2016.

Neil Graham, Graeme Hirst, and Bhaskara Marthi. Segmenting documents by stylistic character. *Natural Language Engineering*, 11(04):397–415, 2005.

Efstathios Stamatatos. A survey of modern authorship attribution methods. *Journal of the American Society for information Science and Technology*, 60(3):538–556, 2009a.

Sven Meyer Zu Eissen and Benno Stein. Intrinsic plagiarism detection. In *European Conference on Information Retrieval*, pages 565–569. Springer, 2006.

Thomas Corwin Mendenhall. The characteristic curves of composition. *Science*, pages 237–249, 1887.

George Kingsley Zipf. Selected studies of the principle of relative frequency in language. 1932.

G Udny Yule. On sentence-length as a statistical characteristic of style in prose: With application to two cases of disputed authorship. *Biometrika*, 30(3/4):363–390, 1939.

William J Backoff, Harry L Moir, John F O'loughlin, Norman D Williams, and John S Yule. Motor fuel composition, May 30 1944. US Patent 2,350,145.

Frederick Mosteller and David Wallace. Inference and disputed authorship: The federalist. 1964.

John Burrows et al. 'delta': a measure of stylistic difference and a guide to likely authorship— nova. the university of newcastle's digital repository. 2002.

David L Hoover. Testing burrows's delta. *Literary and linguistic computing*, 19(4): 453–475, 2004.

Olivier De Vel, Alison Anderson, Malcolm Corney, and George Mohay. Mining e-mail content for author identification forensics. *ACM Sigmod Record*, 30(4):55–64, 2001.

Dominique Estival, Tanja Gaustad, Son Bao Pham, Will Radford, and Ben Hutchinson. Author profiling for english emails. In *Proceedings of the 10th Conference of the Pacific Association for Computational Linguistics (PACLING07)*, pages 263–272, 2007.

Farkhund Iqbal, Hamad Binsalleeh, Benjamin CM Fung, and Mourad Debbabi. Mining writeprints from anonymous e-mails for forensic investigation. *digital investigation*, 7 (1):56–64, 2010.

Ahmed Abbasi and Hsinchun Chen. Applying authorship analysis to extremist-group web forum messages. *IEEE Intelligent Systems*, 20(5):67–75, 2005.

Ahmed Abbasi, Hsinchun Chen, and Arab Salem. Sentiment analysis in multiple languages: Feature selection for opinion classification in web forums. *ACM Transactions on Information Systems (TOIS)*, 26(3):12, 2008.

Thamar Solorio, Sangita Pillay, Sindhu Raghavan, and Manuel Montes-y Gómez. Modality specific meta features for authorship attribution in web forum posts. In *IJCNLP*, pages 156–164, 2011.

Moshe Koppel, Jonathan Schler, Shlomo Argamon, and Eran Messeri. Authorship attribution with thousands of candidate authors. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 659–660. ACM, 2006.

Moshe Koppel, Jonathan Schler, and Shlomo Argamon. Authorship attribution in the wild. *Language Resources and Evaluation*, 45(1):83–94, 2011b.

Tayfun Kucukyilmaz, B Barla Cambazoglu, Cevdet Aykanat, and Fazli Can. Chat mining for gender prediction. In *International Conference on Advances in Information Systems*, pages 274–283. Springer, 2006.

Robert Layton, Paul Watters, and Richard Dazeley. Authorship attribution for twitter in 140 characters or less. In *Cybercrime and Trustworthy Computing Workshop (CTC), 2010 Second*, pages 1–8. IEEE, 2010.

Farkhund Iqbal, Hamad Binsalleeh, Benjamin CM Fung, and Mourad Debbabi. A unified data mining solution for authorship analysis in anonymous textual communications. *Information Sciences*, 231:98–112, 2013.

Ivan Krsul and Eugene H Spafford. Authorship analysis: Identifying the author of a program. *Computers & Security*, 16(3):233–257, 1997.

Steven Burrows, Alexandra L Uitdenbogerd, and Andrew Turpin. Comparing techniques for authorship attribution of source code. *Software: Practice and Experience*, 44(1): 1–32, 2014.

Mamoun Alazab. Profiling and classifying the behavior of malicious codes. *Journal of Systems and Software*, 100:91–102, 2015.

Jane C Ginsburg. The concept of authorship in comparative copyright law. *DePaul L. Rev.*, 52:1063, 2002.

Mohsen Ghasemi Ariani, Fatemeh Sajedi, and Mahin Sajedi. Forensic linguistics: A brief overview of the key elements. *Procedia-Social and Behavioral Sciences*, 158:222–225, 2014.

Tim Grant. Quantifying evidence in forensic authorship analysis. *International Journal of Speech, Language & the Law*, 14(1), 2007.

Rong Zheng, Yi Qin, Zan Huang, and Hsinchun Chen. Authorship analysis in cybercrime investigation. In *International Conference on Intelligence and Security Informatics*, pages 59–73. Springer, 2003.

Rong Zheng, Jiexun Li, Hsinchun Chen, and Zan Huang. A framework for authorship identification of online messages: Writing-style features and classification techniques. *Journal of the American society for information science and technology*, 57(3):378–393, 2006.

Jay Kothari, Maxim Shevertalov, Edward Stehle, and Spiros Mancoridis. A probabilistic approach to source code authorship identification. In *Information Technology, 2007. ITNG'07. Fourth International Conference on*, pages 243–248. IEEE, 2007.

Upul Bandara and Gamini Wijayarathna. Deep neural networks for source code author identification. In *International Conference on Neural Information Processing*, pages 368–375. Springer, 2013.

Martin Potthast, Benno Stein, Alberto Barrón-Cedeño, and Paolo Rosso. An evaluation framework for plagiarism detection. In *Proceedings of the 23rd international conference on computational linguistics: Posters*, pages 997–1005. Association for Computational Linguistics, 2010.

OM Mirza and Mike Joy. Style analysis for source code plagiarism detection. In *Plagiarism Across Europe and Beyond 2015: Conference Proceedings*, pages 53–61, 2015.

Kazushi Ikeda, Gen Hattori, Chihiro Ono, Hideki Asoh, and Teruo Higashino. Twitter user profiling based on text and community mining for market analysis. *Knowledge-Based Systems*, 51:35–47, 2013.

He Jiang, Jingxuan Zhang, Hongjing Ma, Najam Nazar, and Zhilei Ren. Mining authorship characteristics in bug repositories. *Science China Information Sciences*, 2015.

Bei Yu. Function words for chinese authorship attribution. In *CLfL@ NAACL-HLT*, pages 45–53, 2012.

George K Mikros and Eleni K Argiri. Investigating topic influence in authorship attribution. In *PAN*, 2007.

Fuchun Peng, Dale Schuurmans, Shaojun Wang, and Vlado Keselj. Language independent authorship attribution using character level language models. In *Proceedings of the tenth conference on European chapter of the Association for Computational Linguistics-Volume 1*, pages 267–274. Association for Computational Linguistics, 2003.

Vlado Keselj, Fuchun Peng, Nick Cercone, and Calvin Thomas. N-gram-based author profiles for authorship attribution. In *Proceedings of the conference pacific association for computational linguistics, PACLING*, volume 3, pages 255–264, 2003.

Patrick Juola. Ad-hoc authorship attribution competition. In *Proceedings of the Joint Conference of the Association for Computers and the Humanities and the Association for Literary and Linguistic Computing*, pages 175–176. Goteborg, Sweden, 2004.

Maciej Eder. Does size matter? authorship attribution, small samples, big problem. *Digital Scholarship in the Humanities*, page fqt066, 2013.

Moshe Koppel, Jonathan Schler, and Shlomo Argamon. Authorship attribution: What's easy and what's hard? 2013.

Amy Brand, Liz Allen, Micah Altman, Marjorie Hlava, and Jo Scott. Beyond authorship: attribution, contribution, collaboration, and credit. *Learned Publishing*, 28(2):151–155, 2015.

Anderson Rocha, Walter J Scheirer, Christopher W Forstall, Thiago Cavalcante, Antonio Theophilo, Bingyu Shen, Ariadne RB Carvalho, and Efstathios Stamatatos. Authorship attribution for social media forensics. *IEEE Transactions on Information Forensics and Security*, 12(1):5–33, 2017.

Joachim Diederich, Jörg Kindermann, Edda Leopold, and Gerhard Paass. Authorship attribution with support vector machines. *Applied intelligence*, 19(1-2):109–123, 2003.

Mahmoud Khonji, Youssef Iraqi, and Andrew Jones. An evaluation of authorship attribution using random forests. In *Information and Communication Technology Research (ICTRC), 2015 International Conference on*, pages 68–71. IEEE, 2015.

G Bruce Schaalje, Paul J Fields, Matthew Roper, and Gregory L Snow. Extended nearest shrunken centroid classification: a new method for open-set authorship attribution of texts of varying sizes. *Literary and Linguistic Computing*, 26(1):71–88, 2011.

G Bruce Schaalje, Natalie J Blades, and Tomohiko Funai. An open-set size-adjusted bayesian classifier for authorship attribution. *Journal of the American Society for Information Science and Technology*, 64(9):1815–1825, 2013.

Roy Schwartz Oren Tsur Ari Rappoport and Moshe Koppel. Authorship attribution of micro-messages. 2013.

Smita Nirkhi and Rajiv V Dharaskar. Comparative study of authorship identification techniques for cyber forensics analysis. *arXiv preprint arXiv:1401.6118*, 2013.

Kim Luyckx and Walter Daelemans. Authorship attribution and verification with many authors and limited data. In *Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1*, pages 513–520. Association for Computational Linguistics, 2008.

Moshe Koppel and Jonathan Schler. Authorship verification as a one-class classification problem. In *Proceedings of the twenty-first international conference on Machine learning*, page 62. ACM, 2004.

Xiaoling Chen, Peng Hao, Rajarathnam Chandramouli, and KP Subbalakshmi. Authorship similarity detection from email messages. In *International Workshop on Machine Learning and Data Mining in Pattern Recognition*, pages 375–386. Springer, 2011.

Omar Canales, Vinnie Monaco, Thomas Murphy, Edyta Zych, John Stewart, Charles Tappert Alex Castro, Ola Sotoye, Linda Torres, and Greg Truley. A stylometry system for authenticating students taking online tests. *P. of Student-Faculty Research Day, Ed., CSIS. Pace University*, 2011.

Deepa Gupta, K Vani, and Charan Kamal Singh. Using natural language processing techniques and fuzzy-semantic similarity for automatic external plagiarism detection. In *Advances in Computing, Communications and Informatics (ICACCI, 2014 International Conference on*, pages 2694–2699. IEEE, 2014.

K Vani and Deepa Gupta. Using k-means cluster based techniques in external plagiarism detection. In *Contemporary computing and informatics (IC3I), 2014 international conference on*, pages 1268–1273. IEEE, 2014.

N Riya Ravi, K Vani, and Deepa Gupta. Exploration of fuzzy c means clustering algorithm in external plagiarism detection system. In *Intelligent Systems Technologies and Applications*, pages 127–138. Springer, 2016.

Imene Bensalem, Paolo Rosso, and Salim Chikhi. Intrinsic plagiarism detection using n-gram classes. In *EMNLP*, pages 1459–1464, 2014.

Marcin Kuta and Jacek Kitowski. Optimisation of character n-gram profiles method for intrinsic plagiarism detection. In *International Conference on Artificial Intelligence and Soft Computing*, pages 500–511. Springer, 2014.

Adi Wijaya and Romi Satria Wahono. Two-step cluster based feature discretization of naïve bayes for outlier detection in intrinsic plagiarism detection. *Journal of Intelligent Systems*, 1(1):1–8, 2015.

Shlomo Argamon, Moshe Koppel, James W Pennebaker, and Jonathan Schler. Automatically profiling the author of an anonymous text. *Communications of the ACM*, 52(2):119–123, 2009.

Julio Villena Román and José Carlos González Cristóbal. Daedalus at pan 2014: Guessing tweet author's gender and age. 2014.

Jonathan Schler, Moshe Koppel, Shlomo Argamon, and James W Pennebaker. Effects of age and gender on blogging. In *AAAI spring symposium: Computational approaches to analyzing weblogs*, volume 6, pages 199–205, 2006.

Seifeddine Mechti, Maher Jaoua, Lamia Hadrich Belguith, and Rim Faiz. Machine learning for classifying authors of anonymous tweets, blogs, reviews and social media. *Proceedings of the PAN@ CLEF, Sheffield, England*, 2014.

Malcolm Corney, Olivier De Vel, Alison Anderson, and George Mohay. Gender-preferential text mining of e-mail discourse. In *Computer Security Applications Conference, 2002. Proceedings. 18th Annual*, pages 282–289. IEEE, 2002.

Moshe Koppel, Jonathan Schler, and Kfir Zigdon. Determining an author's native language by mining a text for errors. In *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, pages 624–628. ACM, 2005.

Moshe Koppel, Navot Akiva, Eli Alshech, and Kfir Bar. Automatically classifying documents by ideological and organizational affiliation. In *Intelligence and Security Informatics, 2009. ISI'09. IEEE International Conference on*, pages 176–178. IEEE, 2009a.

Dang Duc Pham, Giang Binh Tran, and Son Bao Pham. Author profiling for vietnamese blogs. In *Asian Language Processing, 2009. IALP'09. International Conference on*, pages 190–194. IEEE, 2009.

Moshe Koppel and Yaron Winter. Determining if two documents are written by the same author. *Journal of the Association for Information Science and Technology*, 65 (1):178–187, 2014.

Efstathios Stamatatos, Walter Daelemans, Ben Verhoeven, Patrick Juola, Aurelio López-López, Martin Potthast, and Benno Stein. Overview of the author identification task at pan 2014. In *CLEF (Working Notes)*, pages 877–897, 2014.

Harald Baayen, Hans Van Halteren, and Fiona Tweedie. Outside the cave of shadows: Using syntactic annotation to enhance authorship attribution. *Literary and Linguistic Computing*, 11(3):121–132, 1996.

Efstathios Stamatatos, Nikos Fakotakis, and George Kokkinakis. Automatic text categorization in terms of genre and author. *Computational linguistics*, 26(4):471–495, 2000.

Efstathios Stamatatos, Nikos Fakotakis, and Georgios Kokkinakis. Computer-based authorship attribution without lexical measures. *Computers and the Humanities*, 35 (2):193–214, 2001.

Michael Gamon. Linguistic correlates of style: authorship classification with deep linguistic analysis features. In *Proceedings of the 20th international conference on Computational Linguistics*, page 611. Association for Computational Linguistics, 2004.

Graeme Hirst and Olga Feiguina. Bigrams of syntactic labels for authorship discrimination of short texts. *Literary and Linguistic Computing*, 22(4):405–417, 2007.

Ying Zhao and Justin Zobel. Searching with style: authorship attribution in classic literature. In *Proceedings of the thirtieth Australasian conference on Computer science-Volume 62*, pages 59–68. Australian Computer Society, Inc., 2007.

Tieyun Qian, Bing Liu, Li Chen, and Zhiyong Peng. Tri-training for authorship attribution with limited training data. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 345–351. Association for Computational Linguistics, 2014. URL http://aclweb.org/anthology/P14-2057.

Moshe Koppel and Jonathan Schler. Exploiting stylistic idiosyncrasies for authorship attribution. In *Proceedings of IJCAI'03 Workshop on Computational Approaches to Style Analysis and Synthesis*, volume 69, pages 72–80, 2003.

Shlomo Argamon, Marin Šarić, and Sterling S Stein. Style mining of electronic messages for multiple authorship discrimination: first results. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 475–480. ACM, 2003.

Shlomo Argamon, Casey Whitelaw, Paul Chase, Sobhan Raj Hota, Navendu Garg, and Shlomo Levitan. Stylistic text classification using functional lexical features. *Journal of the American Society for Information Science and Technology*, 58(6):802–822, 2007.

John F Burrows. Word-patterns and story-shapes: The statistical analysis of narrative style. *Literary and linguistic Computing*, 2(2):61–70, 1987.

Shlomo Argamon and Shlomo Levitan. Measuring the usefulness of function words for authorship attribution. In *ACH/ALLC*, 2005.

Jacques Savoy. Authorship attribution based on a probabilistic topic model. *Information Processing & Management*, 49(1):341–354, 2013a.

Jacques Savoy. The federalist papers revisited: A collaborative attribution scheme. *Proceedings of the American Society for Information Science and Technology*, 50(1): 1–8, 2013b.

Giorgio Roffo, Cristina Segalin, Alessandro Vinciarelli, Vittorio Murino, and Marco Cristani. Reading between the turns: Statistical modeling for identity recognition and verification in chats. In *Advanced Video and Signal Based Surveillance (AVSS), 2013 10th IEEE International Conference on*, pages 99–104. IEEE, 2013.

Ambika Shrestha Chitrakar and Katrin Franke. Author identification from text-based communications: Identifying generalized features and computational methods. *Norsk informasjonssikkerhetskonferanse (NISK)*, 2013, 2014.

Santiago Segarra, Mark Eisen, and Alejandro Ribeiro. Authorship attribution through function word adjacency networks. *arXiv preprint arXiv:1406.4469*, 2014.

Conrad Mascol. Curves of pauline and pseudo-pauline style i. *Unitarian Review*, 30: 453–460, 1888.

Claude S Brinegar. Mark twain and the quintus curtius snodgrass letters: A statistical test of authorship. *Journal of the American Statistical Association*, 58(301):85–96, 1963.

G Udny Yule. The statistical study of literary vocabulary, 1946.

Andrew Q Morton. The authorship of greek prose. *Journal of the Royal Statistical Society. Series A (General)*, 128(2):169–233, 1965.

HS Sichel. Word frequency distributions and type-token characteristics. *Mathematical Scientist*, 11(1):45–72, 1986.

Matthew L Jockers and Daniela M Witten. A comparative study of machine learning methods for authorship attribution. *Literary and Linguistic Computing*, page fqq001, 2010.

Bradley Kjell, W Addison Woods, and Ophir Frieder. Discrimination of authorship using visualization. *Information Processing & Management*, 30(1):141–150, 1994.

Jack Grieve. Quantitative authorship attribution: An evaluation of techniques. *Literary and linguistic computing*, 22(3):251–270, 2007.

Johan F Hoorn, Stefan L Frank, Wojtek Kowalczyk, and Floor van Der Ham. Neural network identification of poets using letter sequences. *Literary and Linguistic Computing*, 14(3):311–338, 1999.

Ross Clement and David Sharp. Ngram and bayesian classification of documents for topic and authorship. *Literary and linguistic computing*, 18(4):423–447, 2003.

Ying Zhao and Justin Zobel. Effective and scalable authorship attribution using function words. In *Asia Information Retrieval Symposium*, pages 174–189. Springer, 2005.

Alaa Saleh Altheneyan and Mohamed El Bachir Menai. Naïve bayes classifiers for authorship attribution of arabic texts. *Journal of King Saud University-Computer and Information Sciences*, 26(4):473–484, 2014.

Olivier De Vel. Mining e-mail authorship. In *Proc. Workshop on Text Mining, ACM International Conference on Knowledge Discovery and Data Mining (KDD2000)*, 2000.

Tony Abou-Assaleh, Nick Cercone, Vlado Keselj, and Ray Sweidan. Detection of new malicious code using n-grams signatures. In *PST*, pages 193–196, 2004.

Oren Halvani, Martin Steinebach, and Ralf Zimmermann. Authorship verification via k-nearest neighbor estimation. In *Working Notes for CLEF 2013 Conference, Valencia, Spain*, number 1179, 2013.

Na Cheng, Rajarathnam Chandramouli, and KP Subbalakshmi. Author gender identification from text. *Digital Investigation*, 8(1):78–88, 2011.

David Madigan, Alexander Genkin, David D Lewis, Shlomo Argamon, Dmitriy Fradkin, and Li Ye. Author identification on the large scale. In *Proc. of the Meeting of the Classification Society of North America*, page 13, 2005.

Marius Popescu and Cristian Grozea. Kernel methods and string kernels for authorship analysis. In *CLEF (Online Working Notes/Labs/Workshop)*, 2012.

Lisa Pearl and Mark Steyvers. Detecting authorship deception: a supervised machine learning approach using author writeprints. *Literary and linguistic computing*, page fqs003, 2012.

Jose Hurtado, Napat Taweewitchakreeya, and Xingquan Zhu. Who wrote this paper? learning for authorship de-identification using stylometric featuress. In *Information reuse and integration (IRI), 2014 IEEE 15th international conference on*, pages 859–862. IEEE, 2014.

Robert Layton, Paul Watters, and Richard Dazeley. Automated unsupervised authorship analysis using evidence accumulation clustering. *Natural Language Engineering*, 19 (01):95–120, 2013.

Sotiris B Kotsiantis, I Zaharakis, and P Pintelas. Supervised machine learning: A review of classification techniques, 2007.

George James Lidstone. Note on the general case of the bayes-laplace formula for inductive or a posteriori probabilities. *Transactions of the Faculty of Actuaries*, 8(182-192): 13, 1920.

William Ernest Johnson. I.probability: The deductive and inductive problems. *Mind*, 41(164):409–423, 1932.

Christopher D Manning, Prabhakar Raghavan, Hinrich Schütze, et al. *Introduction to information retrieval*, volume 1. Cambridge university press Cambridge, 2008a.

James H Martin and Daniel Jurafsky. Speech and language processing. *International Edition*, 710, 2000.

Hussein Hazimeh and ChengXiang Zhai. Axiomatic analysis of smoothing methods in language models for pseudo-relevance feedback. In *Proceedings of the 2015 International Conference on The Theory of Information Retrieval*, pages 141–150. ACM, 2015.

Thorsten Joachims. Text categorization with support vector machines: Learning with many relevant features. *Machine learning: ECML-98*, pages 137–142, 1998.

Simon Tong and Daphne Koller. Support vector machine active learning with applications to text classification. *Journal of machine learning research*, 2(Nov):45–66, 2001.

Vito DOrazio, Steven T Landis, Glenn Palmer, and Philip Schrodt. Separating the wheat from the chaff: Applications of automated document classification using support vector machines. *Political analysis*, 22(2), 2014.

Yong H Li and Anil K Jain. Classification of text documents. *The Computer Journal*, 41(8):537–546, 1998.

David E Johnson, Frank J Oles, and Tong Zhang. Decision-tree-based symbolic rule induction system for text categorization, February 11 2003. US Patent 6,519,580.

Dewan Md Farid, Li Zhang, Chowdhury Mofizur Rahman, M Alamgir Hossain, and Rebecca Strachan. Hybrid decision tree and naïve bayes classifiers for multi-class classification tasks. *Expert Systems with Applications*, 41(4):1937–1946, 2014.

Andrew McCallum, Kamal Nigam, et al. A comparison of event models for naive bayes text classification. In *AAAI-98 workshop on learning for text categorization*, volume 752, pages 41–48. Citeseer, 1998.

Ioan Pop. An approach of the naive bayes classifier for the document classification. *General Mathematics*, 14(4):135–138, 2006.

SL Ting, WH Ip, and Albert HC Tsang. Is naive bayes a good classifier for document classification. *International Journal of Software Engineering and Its Applications*, 5 (3):37–46, 2011.

Eui-Hong Sam Han, George Karypis, and Vipin Kumar. Text categorization using weight adjusted k-nearest neighbor classification. In *Pacific-asia conference on knowledge discovery and data mining*, pages 53–65. Springer, 2001.

Min-Ling Zhang and Zhi-Hua Zhou. A k-nearest neighbor based algorithm for multi-label classification. In *Granular Computing, 2005 IEEE International Conference on*, volume 2, pages 718–721. IEEE, 2005.

Shengyi Jiang, Guansong Pang, Meiling Wu, and Limin Kuang. An improved k-nearest-neighbor algorithm for text categorization. *Expert Systems with Applications*, 39(1): 1503–1509, 2012.

Jennifer Farkas. Neural networks and document classification. In *Electrical and Computer Engineering, 1993. Canadian Conference on*, pages 1–4. IEEE, 1993.

Larry Manevitz and Malik Yousef. One-class document classification via neural networks. *Neurocomputing*, 70(7):1466–1481, 2007.

Rodrigo Moraes, JoãO Francisco Valiati, and Wilson P GaviãO Neto. Document-level sentiment classification: An empirical comparison between svm and ann. *Expert Systems with Applications*, 40(2):621–633, 2013.

Kamal Nigam, John Lafferty, and Andrew McCallum. Using maximum entropy for text classification. In *IJCAI-99 workshop on machine learning for information filtering*, volume 1, pages 61–67, 1999.

Shenghuo Zhu, Xiang Ji, Wei Xu, and Yihong Gong. Multi-labelled classification using maximum entropy method. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 274–281. ACM, 2005.

Alaa M El-Halees. Arabic text classification using maximum entropy. *IUG Journal of Natural Studies*, 15(1), 2015.

Qiang Ye, Ziqiong Zhang, and Rob Law. Sentiment classification of online reviews to travel destinations by supervised machine learning approaches. *Expert Systems with Applications*, 36(3):6527–6535, 2009.

Mohammed Abdul Wajeed and T Adilakshmi. Semi-supervised text classification using enhanced knn algorithm. In *Information and Communication Technologies (WICT), 2011 World Congress on*, pages 138–142. IEEE, 2011.

Yuguang Huang and Lei Li. Naive bayes classification algorithm based on small sample set. In *Cloud Computing and Intelligence Systems (CCIS), 2011 IEEE International Conference on*, pages 34–39. IEEE, 2011.

MF Hussin and M Kamel. Document clustering using hierarchical somart neural network. In *Neural Networks, 2003. Proceedings of the International Joint Conference on*, volume 3, pages 2238–2242. IEEE, 2003.

Nicoleta Rogovschi, Mustapha Lebbah, and Younes Bennani. Learning self-organizing mixture markov models. *Journal of Nonlinear Systems and Applications*, 1:63–71, 2010.

Stephen E Levinson, Lawrence R Rabiner, and Man Mohan Sondhi. An introduction to the application of the theory of probabilistic functions of a markov process to automatic speech recognition. *The Bell System Technical Journal*, 62(4):1035–1074, 1983.

Biing Hwang Juang and Laurence R Rabiner. Hidden markov models for speech recognition. *Technometrics*, 33(3):251–272, 1991.

Junji Yamato, Jun Ohya, and Kenichiro Ishii. Recognizing human action in time-sequential images using hidden markov model. In *Computer Vision and Pattern Recognition, 1992. Proceedings CVPR'92., 1992 IEEE Computer Society Conference on*, pages 379–385. IEEE, 1992.

Mohiuddin Ahmad and Seong-Whan Lee. Hmm-based human action recognition using multiview image sequences. In *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*, volume 1, pages 263–266. IEEE, 2006.

Jianying Hu, Michael K Brown, and William Turin. Hmm based online handwriting recognition. *IEEE Transactions on pattern analysis and machine intelligence*, 18(10): 1039–1045, 1996.

U-V Marti and Horst Bunke. Using a statistical language model to improve the performance of an hmm-based cursive handwriting recognition system. *International journal of Pattern Recognition and Artificial intelligence*, 15(01):65–90, 2001.

Thorsten Brants. Tnt: a statistical part-of-speech tagger. In *Proceedings of the sixth conference on Applied natural language processing*, pages 224–231. Association for Computational Linguistics, 2000.

Michael Collins. Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*, pages 1–8. Association for Computational Linguistics, 2002.

Ara V Nefian and Monson H Hayes. Hidden markov models for face recognition. In *Acoustics, Speech and Signal Processing, 1998. Proceedings of the 1998 IEEE International Conference on*, volume 5, pages 2721–2724. IEEE, 1998.

Xiaoming Liu and Tsuhan Cheng. Video-based face recognition using adaptive hidden markov models. In *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, volume 1, pages I–I. IEEE, 2003.

Lawrence R Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society. Series B (methodological)*, pages 1–38, 1977.

Lawrence R Rabiner and Biing-Hwang Juang. An introduction to hidden markov models. *ASSP Magazine, IEEE*, 3(1):4–16, 1986.

Richard O Duda, Peter E Hart, and David G Stork. Pattern classification. 2nd. *Edition. New York*, 2001.

Andrew J Viterbi. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *Information Theory, IEEE Transactions on*, 13(2):260–269, 1967.

Glenn A Bowen. Document analysis as a qualitative research method. *Qualitative research journal*, 9(2):27–40, 2009.

Julian Kupiec. Robust part-of-speech tagging using a hidden markov model. *Computer Speech & Language*, 6(3):225–242, 1992.

Karl Stratos, Michael Collins, and Daniel Hsu. Unsupervised part-of-speech tagging with anchor hidden markov models. *Transactions of the Association for Computational Linguistics*, 4:245–257, 2016.

Scott M Thede and Mary P Harper. A second-order hidden markov model for part-of-speech tagging. In *Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics*, pages 175–182. Association for Computational Linguistics, 1999.

Jonathan P Yamron, Ira Carp, Larry Gillick, Steve Lowe, and Paul van Mulbregt. A hidden markov model approach to text segmentation and event tracking. In *Acoustics, Speech and Signal Processing, 1998. Proceedings of the 1998 IEEE International Conference on*, volume 1, pages 333–336. IEEE, 1998.

David M Blei and Pedro J Moreno. Topic segmentation with an aspect hidden markov model. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 343–348. ACM, 2001.

Ludovic Denoyer, Hugo Zaragoza, and Patrick Gallinari. Hmm-based passage models for document classification and ranking. In *Proceedings of ECIR-01, 23rd European Colloquium on Information Retrieval Research*, pages 126–135, 2001.

Tsimboukakis Nikolaos and Tambouratzis George. Document classification system based on hmm word map. In *Proceedings of the 5th international conference on Soft computing as transdisciplinary science and technology*, pages 7–12. ACM, 2008.

Kwan Yi and Jamshid Beheshti. A text categorization model based on hidden markov models. In *Proceedings of the Annual Conference of CAIS/Actes du congrès annuel de l'ACSI*, 2013.

A Seara Vieira, Eva Lorenzo Iglesias, and Lourdes Borrajo. T-hmm: a novel biomedical text classifier based on hidden markov models. In *8th International Conference on Practical Applications of Computational Biology & Bioinformatics (PACBB 2014)*, pages 225–234. Springer, 2014.

Yi-Ting Chen, Suhan Yu, Hsin-Min Wang, and Berlin Chen. Extractive chinese spoken document summarization using probabilistic ranking models. In *Chinese Spoken Language Processing*, pages 660–671. Springer, 2006.

Sameer Maskey and Julia Hirschberg. Summarizing speech without text using hidden markov models. In *Proceedings of the Human Language Technology Conference of the*

NAACL, Companion Volume: Short Papers, pages 89–92. Association for Computational Linguistics, 2006.

David Pinto, Andrew McCallum, Xing Wei, and W Bruce Croft. Table extraction using conditional random fields. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 235–242. ACM, 2003.

Ana Costa e Silva. Learning rich hidden markov models in document analysis: Table location. In *Document Analysis and Recognition, 2009. ICDAR'09. 10th International Conference on*, pages 843–847. IEEE, 2009.

Vilaythong Southavilay, Kalina Yacef, and Rafael A Calvo. Analysis of collaborative writing processes using hidden markov models and semantic heuristics. In *Data Mining Workshops (ICDMW), 2010 IEEE International Conference on*, pages 543–548. IEEE, 2010.

James MacQueen et al. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297. Oakland, CA, USA., 1967.

Ulrike Von Luxburg. A tutorial on spectral clustering. *Statistics and computing*, 17(4): 395–416, 2007.

James C Bezdek, Robert Ehrlich, and William Full. Fcm: The fuzzy c-means clustering algorithm. *Computers & Geosciences*, 10(2-3):191–203, 1984.

Asoke K Nandi and Detlef Mämpel. An extension of the generalized gaussian distribution to include asymmetry. *Journal of the Franklin Institute*, 332(1):67–75, 1995.

G Suvarna Kumar, KA PRASAD Raju, Mohan Rao CPVNJ, and P Satheesh. Speaker recognition using gmm. *International Journal of Engineering Science and Technology*, 2(6):2428–2436, 2010.

Jeff A Bilmes et al. A gentle tutorial of the em algorithm and its application to parameter estimation for gaussian mixture and hidden markov models. *International Computer Science Institute*, 4(510):126, 1998.

Christopher M Bishop. Pattern recognition. *Machine Learning*, 128:1–58, 2006.

Yang Xian, Xiaodong Yang, and Yingli Tian. Hybrid example-based single image super-resolution. In *International Symposium on Visual Computing*, pages 3–15. Springer, 2015.

Xin Liu, Yihong Gong, Wei Xu, and Shenghuo Zhu. Document clustering with cluster refinement and model selection capabilities. In *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 191–198. ACM, 2002.

Chao Xing, Dong Wang, Xuewei Zhang, and Chao Liu. Document classification with distributions of word vectors. In *Asia-Pacific Signal and Information Processing Association, 2014 Annual Summit and Conference (APSIPA)*, pages 1–5. IEEE, 2014.

Andreas Schlapbach and Horst Bunke. Off-linewriter identification using gaussian mixture models. In *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*, volume 3, pages 992–995. IEEE, 2006.

Andreas Schlapbach and Horst Bunke. Off-line writer identification and verification using gaussian mixture models. In *Machine learning in document analysis and recognition*, pages 409–428. Springer, 2008.

Vincent Christlein, David Bernecker, Florian Honig, and Elli Angelopoulou. Writer identification and verification using gmm supervectors. In *Applications of Computer Vision (WACV), 2014 IEEE Winter Conference on*, pages 998–1005. IEEE, 2014.

Freddy YY Choi. Advances in domain independent linear text segmentation. In *Proceedings of the 1st North American chapter of the Association for Computational Linguistics conference*, pages 26–33. Association for Computational Linguistics, 2000.

Hemant Misra, François Yvon, Joemon M Jose, and Olivier Cappe. Text segmentation via topic modeling: an analytical study. In *Proceedings of the 18th ACM conference on Information and knowledge management*, pages 1553–1556. ACM, 2009.

Farkhund Iqbal, Rachid Hadjidj, Benjamin CM Fung, and Mourad Debbabi. A novel approach of mining write-prints for authorship attribution in e-mail forensics. *digital investigation*, 5:S42–S51, 2008.

Sven Meyer Zu Eissen, Benno Stein, and Marion Kulig. Plagiarism detection without reference collections. In *Advances in data analysis*, pages 359–366. Springer, 2007.

Efstathios Stamatatos. Intrinsic plagiarism detection using character n-gram profiles. *threshold*, 2:1–500, 2009b.

Michal Rosen-Zvi, Thomas Griffiths, Mark Steyvers, and Padhraic Smyth. The author-topic model for authors and documents. In *Proceedings of the 20th conference on Uncertainty in artificial intelligence*, pages 487–494. AUAI Press, 2004.

Michal Rosen-Zvi, Chaitanya Chemudugunta, Thomas Griffiths, Padhraic Smyth, and Mark Steyvers. Learning author-topic models from text corpora. *ACM Transactions on Information Systems (TOIS)*, 28(1):4, 2010.

Ying Zhao and George Karypis. Criterion functions for document clustering: Experiments and analysis. 2001.

Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA, 2008b. ISBN 0521865719, 9780521865715.

Enrique Amigó, Julio Gonzalo, Javier Artiles, and Felisa Verdejo. A comparison of extrinsic clustering evaluation metrics based on formal constraints. *Information Retrieval*, 12(4):461–486, 2009.

Harr Chen, SRK Branavan, Regina Barzilay, David R Karger, et al. Content modeling using latent permutations. *Journal of Artificial Intelligence Research*, 36(1):129–163, 2009.

Shafiq Joty, Giuseppe Carenini, and Raymond T Ng. Topic segmentation and labeling in asynchronous conversations. *Journal of Artificial Intelligence Research*, pages 521–573, 2013.

Thomas Hofmann. Probabilistic latent semantic indexing. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 50–57. ACM, 1999.

Bo Han, Paul Cook, and Timothy Baldwin. Text-based twitter user geolocation prediction. *Journal of Artificial Intelligence Research*, pages 451–500, 2014.

Fuchun Peng, Dale Schuurmans, and Shaojun Wang. Augmenting naive bayes classifiers with statistical language models. *Information Retrieval*, 7(3-4):317–345, 2004.

Ahmed Abbasi and Hsinchun Chen. Writeprints: A stylometric approach to identity-level identification and similarity detection in cyberspace. *ACM Trans. Inf. Syst.*, 26 (2):7:1–7:29, April 2008.

Efstathios Stamatatos. Author identification: Using text sampling to handle the class imbalance problem. *Information Processing & Management*, 44(2):790–799, 2008.

Moshe Koppel, Jonathan Schler, and Shlomo Argamon. Computational methods in authorship attribution. *Journal of the American Society for Information Science and Technology*, 60(1):9–26, 2009b.

Merran Evans, Nicholas Hastings, and Brian Peacock. Statistical distributions. 2000.

Marcelo Luiz Brocardo, Issa Traore, Shatina Saad, and Isaac Woungang. Authorship verification for short messages using stylometry. In *Computer, Information and Telecommunication Systems (CITS), 2013 International Conference on*, pages 1–6. IEEE, 2013.

Nektaria Potha and Efstathios Stamatatos. A profile-based method for authorship verification. In *Artificial Intelligence: Methods and Applications*, pages 313–326. Springer, 2014.

Benno Stein, Nedim Lipka, and Peter Prettenhofer. Intrinsic plagiarism analysis. *Language Resources and Evaluation*, 45(1):63–82, 2011.

Mike Kestemont, Kim Luyckx, and Walter Daelemans. Intrinsic plagiarism detection using character trigram distance scores. *Proceedings of the PAN*, 2011.

Patrick Juola. Authorship attribution. *Foundations and Trends in information Retrieval*, 1(3):233–334, 2006.

Jacques Savoy. Estimating the probability of an authorship attribution. *Journal of the Association for Information Science and Technology*, 6(67):1462–1472, 2016.

Ossama Abdel-Hamid, Abdel-rahman Mohamed, Hui Jiang, and Gerald Penn. Applying convolutional neural networks concepts to hybrid NN-HMM model for speech recognition. In *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*, pages 4277–4280. IEEE, 2012.

Mohamed Debyeche, Jean Paul Haton, and Amrane Houacine. A new vector quantization approach for discrete hmm speech recognition system. *International Journal of Computing*, 5(1):72–78, 2014.

Salvador Espana-Boquera, Maria Jose Castro-Bleda, Jorge Gorbe-Moya, and Franisco Zamora-Martinez. Improving offline handwritten text recognition with hybrid hmm/ann models. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(4):767–779, 2011.

Sandip Roy, Partha Pratim Roy, Palaiahnakote Shivakumara, Georgios Louloudis, Chew Lim Tan, and Umapada Pal. Hmm-based multi oriented text recognition in natural scene image. In *Pattern Recognition (ACPR), 2013 2nd IAPR Asian Conference on*, pages 288–292. IEEE, 2013.

Pierre Baldi and Søren Brunak. *Bioinformatics: the machine learning approach*. MIT press, 2001.

Travis J Wheeler, Jody Clements, Sean R Eddy, Robert Hubley, Thomas A Jones, Jerzy Jurka, Arian FA Smit, and Robert D Finn. Dfam: a database of repetitive DNA based on profile hidden Markov models. *Nucleic acids research*, 41(D1):D70–D82, 2013.

Leonard E Baum. An equality and associated maximization technique in statistical estimation for probabilistic functions of Markov processes. *Inequalities*, 3:1–8, 1972.

G David Forney Jr. The Viterbi algorithm. *Proceedings of the IEEE*, 61(3):268–278, 1973.

CF Jeff Wu. On the convergence properties of the em algorithm. *The Annals of statistics*, pages 95–103, 1983.

Lei Xu and Michael I Jordan. On convergence properties of the em algorithm for gaussian mixtures. *Neural computation*, 8(1):129–151, 1996.

Md Shamsul Huda, Ranadhir Ghosh, and John Yearwood. A variable initialization approach to the EM algorithm for better estimation of the parameters of hidden Markov model based acoustic modeling of speech signals. In *Advances in Data Mining. Applications in Medicine, Web Mining, Marketing, Image and Signal Mining*, pages 416–430. Springer, 2006.

Geoffrey McLachlan and David Peel. *Finite mixture models*. John Wiley & Sons, 2004.

Harald Baayen, Hans van Halteren, Anneke Neijt, and Fiona Tweedie. An experiment in authorship attribution. In *6th JADT*, pages 29–37. Citeseer, 2002.

Pauline J Sheldon. An authorship analysis of tourism research. *Annals of Tourism Research*, 18(3):473–484, 1991.

David I Holmes and Richard S Forsyth. The Federalist revisited: New directions in authorship attribution. *Literary and Linguistic Computing*, 10(2):111–127, 1995.

David I Holmes. The evolution of stylometry in humanities scholarship. *Literary and linguistic computing*, 13(3):111–117, 1998.

John M McDowell and Michael Melvin. The determinants of co-authorship: An analysis of the economics literature. *The review of economics and statistics*, pages 155–160, 1983.

SM Nirkhi, Rajiv V Dharaskar, and VM Thakre. Analysis of online messages for identity tracing in cybercrime investigation. In *Cyber Security, Cyber Warfare and Digital Forensic (CyberSec), 2012 International Conference on*, pages 300–305. IEEE, 2012.

Ivan Krsul and Eugene H Spafford. Authorship analysis: Identifying the author of a program. In *Proceedings of the Eighteenth National Information Systems Security Conference*, pages 514–524, 1995.

Nathan Rosenblum, Xiaojin Zhu, and Barton P Miller. Who wrote this code? identifying the authors of program binaries. In *Computer Security–ESORICS 2011*, pages 172–189. Springer, 2011.

Sara El Manar El and Ismail Kassou. Authorship analysis studies: A survey. *International Journal of Computer Applications*, 86(12), 2014.

Kim Luyckx and Walter Daelemans. The effect of author set size and data size in authorship attribution. *Literary and linguistic Computing*, 26(1):35–55, 2011.

Efstathios Stamatatos. Plagiarism detection using stopword n-grams. *Journal of the American Society for Information Science and Technology*, 62(12):2512–2527, 2011.

Angela Orebaugh, Jason Kinser, and Jeremy Allnutt. Visualizing instant messaging author writeprints for forensic analysis. In *Proceedings of the Conference on Digital Forensics, Security and Law*, pages 191–214, 2014.

Doug Beeferman, Adam Berger, and John Lafferty. Statistical models for text segmentation. *Machine Learning*, 34(1-3):177–210, 1999.

James Allan. *Topic detection and tracking: event-based information organization*, volume 12. Springer Science & Business Media, 2012.

Shoaib Jameel and Wai Lam. An unsupervised topic segmentation model incorporating word order. In *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*, pages 203–212. ACM, 2013.

Gabriel Oberreuter, Gaston LHuillier, Sebastian A Rios, and Juan D Velasquez. Approaches for intrinsic and external plagiarism detection. *Proceedings of the PAN*, 2011.

Sameer Rao, Parth Gupta, Khushboo Singhal, and Prasenjit Majumder. External & Intrinsic Plagiarism Detection: VSM & Discourse Markers based Approach Notebook for PAN at CLEF 2011. 2011.

Julian Brooke, Adam Hammond, and Graeme Hirst. Unsupervised stylistic segmentation of poetry with change curves and extrinsic features. In *Proceedings of the NAACL 12 Workshop on Computational Linguistics for Literature. Montreal, QC*, pages 26–35, 2012.

Julian Brooke, Graeme Hirst, and Adam Hammond. Clustering voices in the waste land. In *Proceedings of the 2nd Workshop on Computational Literature for Literature (CLFL13), Atlanta*, 2013.

Thomas G Dietterich. Machine learning for sequential data: A review. In *Structural, Syntactic, and Statistical Pattern Recognition*, pages 15–30. Springer, 2002.

Anders Krogh. Two methods for improving performance of an hmm and their application for gene finding. *Center for Biological Sequence Analysis. Phone*, 45:4525, 1997.

Aditya Gupta and Bhuwan Dhingra. Stock market prediction using hidden Markov models. In *Engineering and Systems (SCES), 2012 Students Conference on*, pages 1–4. IEEE, 2012.

Xueying Zhang, Yiping Wang, and Zhefeng Zhao. A hybrid speech recognition training method for hmm based on genetic algorithm and baum welch algorithm. In *Innovative Computing, Information and Control, 2007. ICICIC'07. Second International Conference on*, pages 572–572. IEEE, 2007.

Xuan Dau Hoang and Jiankun Hu. An efficient hidden Markov model training scheme for anomaly intrusion detection of server applications based on system calls. In *Networks, 2004.(ICON 2004). Proceedings. 12th IEEE International Conference on*, volume 2, pages 470–474. IEEE, 2004.

Geoffrey J McLachlan and Kaye E Basford. *Mixture models: Inference and applications to clustering*, volume 84. Marcel Dekker, 1988.

Christopher D Manning and Hinrich Schütze. *Foundations of statistical natural language processing*. MIT press, 1999.

Khaled Aldebei, Xiangjian He, and Jie Yang. Unsupervised decomposition of a multi-author document based on hidden Markov model. In *Submitted to of the 15th Annual Conference of the North American Chapter of the Association for Computational Linguistics*, 2016b.